MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE: (08) 272 0966

ISSUE 7
JUNE 1980 PRICE $2.50

***** CONTENTS *****

MICRO-80 is registered for posting as a publication - CATEGORY B.

Printed by: The Shovel and Bull Printers,
312A Unley Road, Hyde Park, SA 5061.

All enquiries should be directed to:-
MICRO-80, P.O. Box 231, GOODWOOD, SA 5034
Telephone (08) 272 0966
284 Goodwood Rd., Clarence Park, Adelaide.

** ABOUT MICRO-8Ø **

EDITOR: Ian Vagg

ASSOCIATE EDITORS:

SOFTWARE: Peter Hartley

HARDWARE: Edwin Paay

MICRO-8Ø is the only Australian monthly magazine devoted entirely to the Tandy TRS-8Ø microcomputer and the Dick Smith System 8Ø. It is available by subscription, $24.00 for 12 months or by mail order at $2.50 per copy. A cassette containing all the programs in each month's issue is available for an additional $3.50 or a combined annual subscription to both magazine and cassette is available for $60.00. Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-8Ø is to publish software and other information to help you get the most from your TRS-8Ø or System 8Ø and their peripherals. MICRO-8Ø is in no way connected with either the Tandy or Dick Smith organisations.

** WE WILL PAY YOU TO PUBLISH YOUR PROGRAMS **

Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-8Ø or System 8Ø to earn someextra income is included in every issue.

** CONTENT **

Each month we publish at least one applications program in Level I BASIC, one in Level 2 BASIC and one in DISK BASIC (or disk compatible Level 2). We also publish Utility programs in Level 2 BASIC and Machine Language. At least every second issue has an article on hardware modifications or a constructional article for a useful peripheral. In addition, we run articles on programming techniques both in Assembly Language and BASIC and we print letters to the Editor and new product reviews.

** COPYRIGHT **

All the material published in this magazine is under copyright. That means that you must not copy it, except for your own use. This applies to photocopying the magazine itself or making copies of programs on tape or disk.

** LIABILITY **

The programs and other articles in MICRO-8Ø are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

** EDITORIAL **

The Sydney Home Computer Show was held from May 22 to May 25. MICRO-80 was there nestled between two other well known names in the microcomputing world, TANDY Electronics and DICK SMITH Electronics. From all accounts, the attendance at the Show was down on last year but that is hard to believe from the activity around the MICRO-80 stand. People were often three deep and it was hard for us to give everyone the attention we would have liked. If you were there and didn't get the chance to talk to us, we're sorry, we tried very hard to meet everyone we could.

One of the features of our stand was the System 80 we had on display. Although we had had it for less than a week when the show opened (and a pretty hectic week at that) it was clear that we knew as much about the System 80 as anyone else at the show. There was a lot of interest in the machine and many people asked us our opinion of it. One of these enquirers turned out to be a fairly senior member of the Dick Smith camp. When he revealed himself at a later time, he congratulated us for giving an objective assessment of the System 80 and also told us that Dick Smith's are doing their utmost to have the design revised so that it is EXACTLY the same as the TRS-80. You can read Peter Hartley's review of the System 80 in this issue. There are some differences from the TRS-80 and, unfortunately, much of MICRO-80 PRODUCTS software will not run on the System 80 without some modification. The modifications are being made as a matter of urgency and System 80 owners can rest assured that they too, will be able to obtain as much support from MICRO-80 and MICRO-80 PRODUCTS as TRS-80 owners have been receiving for the past seven months.

It is fairly traditional for computer magazines to present a review of all the exhibitors at shows such as this one. Unfortunately, we were so busy serving the visitors at our stand that we did not have sufficient time to visit the other exhibitors. Therefore, we will leave the reviews to the other magazines.

We did establish a number of contacts which we believe will be of benefit to all of us over the coming months. One member of TANDY's staff took us to task for the "rubbish" we printed in the APRIL issue concerning the future of the TRS-80 versus the "TRS-90". We thought we made it pretty clear that the information published was a collection of conflicting rumours which are being given considerable publicity in the U.S.A. Our final word was of encouragement to TRS-80 owners. Be that as it may, we have given TANDY's spokesman "equal space" to put the record straight, later in these columns. We have also been invited to check out such rumours directly with TANDY Australia before publishing them in future issues. This we will be only too happy to do.

** SPECIAL DISCOUNTS ON TANDY PRODUCTS FOR MICRO-80 READERS **

One of the more exciting outcomes from the Sydney Show is the discount rates for TANDY equipment we have been able to negotiate for MICRO-80 readers. It is common in the U.S.A. for Authorised Radio Shack (Tandy) Dealers to offer TANDY equipment at discounts of about 10%. For some reason, this practice has not yet extended to Australia. At the Show, we met Barry Lawrence of Conquest Electronics, 212 Katoomba St., Katoomba, N.S.W. It didn't take Barry long to agree to our suggestion that it would benefit both MICRO-80 readers and Conquest Electronics if he were to offer similar discounts. Barry has gone one better than most U.S.A. discounters, his discounted prices include postage or freight, so you can save a full 10% on all TANDY equipment and software. Since Conquest Electronics is an authorised TANDY dealer, everything they sell carries TANDY's worldwide guarantee and, should anything go wrong, you can take your purchase into your nearest TANDY store for warranty repairs, just as if you had bought it there. You will see Conquest Electronics' advertisement elsewhere in this issue and we would commend you to support this enterprising organisation and save some money at the same time.

** RUMOUR MILL - TANDY's ANSWER **

As I said earlier, TANDY did not like any suggestions that the TRS-80 Model 1 is to be discontinued. It is true that a new computer which will interface to a colour T.V. is about to be released in the States. TANDY Australia is not sure if this is to be designated the TRS-90 but do say that it is an interim model to counter the TI computer and will not be released in this country because of the difficulty of converting it to work with the PAL television system.

As far as the TRS-80 Model 1 is concerned, that will "go on forever", in all its variants including Level 1 4K. That is clearly an exaggeration but we are relieved to have TANDY's reassurance. Incidentally, speculation continues as to how many TRS-80's have really been produced. TANDY itself advertises "over 150,000 sold", whilst the June edition of the American BYTE magazine reports a top Radio Shack executive as saying that, as at March 1, 1980, Radio Shack had manufactured 370,000 TRS-80's! (Radio Shack is the TANDY subsidiary which manufactures the TRS-80). TANDY are cagey about the number of TRS-80's sold here in Australia. The most persistent figure that crops up is 6000 but some say that only represents the number of Level II machines sold and that there are nearer 10,000 if you also consider the Level I machines sold, of which about 40% are supposed to have been upgraded to Level II. Whatever the figure, the interest shown at the Sydney Show indicates that it is increasing rapidly.

** OUR MEMORY EXPANSION UNIT - GOOD NEWS AND BAD **

First the good news, the prototype of our memory expansion unit is finished and works - we demonstrated it at the Sydney Show. Now for the bad news. We have already invested a considerable amount of money in this unit and much more is needed to produce the double-sided printed circuit board. We need at least 100 firm orders to make this an economic proposition. At the Sydney Show there was so little interest that we have virtually decided to shelve the whole project unless this article elicits a lot of support from our readers. The memory expansion unit is entirely self-contained in a small cabinet about 8" x 10" x 3" high which plugs in to the port on the back of the CPU via a cable which would be supplied in the price. Power comes from the existing TANDY transformer (yes, it has sufficient capacity) which plugs into a DIN socket in the cabinet. A lead from the cabinet plugs into the power socket on the CPU, thus providing it with power. Mounted on the cabinet is an ordinary toggle switch. In one position the memory in the expansion unit starts from 5000 Hex for a 4K CPU and in the other it starts from 8000 Hex, for a 16K CPU. There are sockets on the PC board in the cabinet for up to 32K of memory in two 16K blocks so you can have a 20K, 32K, 36K or 48K machine.

The unit would cost $220 complete but without memory chips. Each 16K of memory would cost about $97 (see MICRO-80 PRODUCTS advertisement). The PC board would include the tracks etc. for an RS 232C serial interface with the same specification as the TANDY unit. To have this part of the board loaded and tested would cost an extra $75. We believe the unit could be made to operate with a System 80 as well as a TRS-80.

If you would really be interested in such a unit, please write in and tell us. Do not send any money yet. One possibility is that we could produce the printed circuit board only and sell it, together with a circuit diagram and instructions for building the unit yourself. In that event, the price would be about $65. If you are more interested in that proposition than the complete unit, please let us know. So, now it's over to our readers.

** MARKET PLACE **

We have become increasingly aware that a number of our readers have acquired their computers second-hand. It is likely, with so many systems in circulation, that there must be quite a few people who have systems for sale and many others who would like to buy them. Therefore, starting next month, we will have a MARKET PLACE column. There will be no charge for advertisements but, on successful completion of a sale, we will require a commission of 5% of the advertised price. Our minimum commission will be $5 and the maximum $30. So, if you advertise a system for $400 say, when you sell it you would pay us a commission of $20. If you advertise a system for $2000 then after the sale, you would pay us our maximum commission of $30. Advertisements should be no longer than 50 words, including address, telephone number, etc. So, if you have a TRS-80 or System 80 or peripheral you would like to sell, send in your advertisement on the form provided. We will print it and you can sit back and wait for the enquiries to roll in. This offer is confined to private individuals selling their own, used equipment only. It does not apply to software.

- 000 -

** SYSTEM 8Ø ** A review by Peter Hartley.

It is now twelve months since the Dick Smith organisation announced the System 8Ø. Ever since, an amazing crop of tales, some tall, some true, have surrounded the new machine. We have been told of "World-wide shortages of memory chips", "Dick Smith and Tandy Corporation have their lawyers at forty paces", "Factories and warehouses have suddenly burned to the ground", and even "Microsoft have gone bankrupt...! "

The System 8Ø is NOW a fact - it IS real - we've touched one! Some things we like - some things we could quickly come to hate! And that is no more, nor any less, than could be said for any other computer on the market.

Before getting into too many specifics, let's make our position quite clear. Micro-8Ø has purchased, over-the-counter, one Dick Smith System 8Ø. Unless we state otherwise, the comments we make relate to this one unit, and may not be generally applicable to every System 8Ø sold, even though it is a mass-produced item.

The System 8Ø arrived in an adequate (just adequate enough to get it home, but not adequate enough to survive being sent to Sydney by carrier if it became unserviceable!) silver -grey box. Inside, the System 8Ø was snuggled up in its polythene bag, well supported by two moulded expanded-polystyrene frames. A separate bag contained cassette connector leads (for the second cassette) and three instruction books were in a third bag.

The visual appearance of the System 8Ø is brilliant. Almost as good as the Sorcerer, and far more attractive than the TRS-8Ø. The mains lead was of sufficient length for most applications, and we had the choice of using one of our Tandy monitors (the System 8Ø doesn't come with its own monitor) or using the inbuilt U.H.F. modulated output and a regular T.V. We took a punt, and plugged in old faithful - wonder of wonders - the pin configuration is Tandy standard! A couple of moments to reset the vertical hold and the screen said...
READY ?
          typing in...
                    produced...
Our first surprise - a blank screen, with another
READY ?
          right at the bottom. Funny thing, that, we thought, and reached for the (CLEAR) key. Surprise number two. We couldn't find it. Above the key labelled (NEW LINE) (which is the same as the (ENTER) key of the TRS-8Ø) was one labelled (bck/spc), but no right-pointing arrow (the TAB function) and no (CLEAR).

A further inspection revealed that the up and down arrow keys, which we have come to expect on the left hand side of the keyboard, were labelled (escp) and (cntrl). Pressing (cntrl) produced a line-feed, just as we thought it should do, but pressing (escp) produced a squared bracket symbol.

Immediately our thoughts were of our growing library of software - and especially of the games that used all four"arrow keys". Submarined!

We then decided to inspect the character set, using the following routine (which would also test whether the video memory used the Tandy addresses)

```
1Ø CLS
2Ø M=1536Ø
3Ø FORK=ØTO255
4Ø POKEM,K
5Ø M=M+2
6Ø NEXT
7Ø PRINT@512,"";
8Ø INPUTD$
9Ø PRINTCHR$(23);
1ØØ GOTO1ØØ
```

The good news is that it worked, but there is one Hell of a lot of bad news. Some parts of the character set weren't TRS-8Ø (generally the useful things like arrows), and line 9Ø appeared not to work at all! However, when we pressed (BREAK) we were greeted with a double spaced (but otherwise normal-sized) READY. This caused a hurried grab for the manuals, and the subsequent discovery of a huge button on the rear of the System 8Ø, labelled (most inappropriately we thought) VIDEO CUT. Pressing this causes the System 8Ø to switch from 64 character mode to 32 character mode, but NOT IN A TANDY COMPATIBLE FORMAT.

When in 32 character mode, the TRS-8Ø displays ALTERNATE bytes of video memory. The System 8Ø, however, displays either the FIRST 32 bytes of each line, or the LAST 32 bytes, according to the position of yet another switch - labelled (PAGE). Printing CHR$(23) on the TRS-8Ø trips a gate, to enable the 32 character format, but on the System 8Ø, the same instruction merely causes the unit to insert a space between each character PRINTED AFTER THE CHR$(23)...

There are some other changes, too. There doesn't appear to be a L3 ERROR message. The Disk Basic Pointers, used in Level II, are addressed to SN ERROR instead. But, importantly, these pointers ARE there, so we are led to the conclusion that the machine will eventually (and we'll discuss "eventually" eventually!) be compatible with TRSDOS and other MICROSOFT DISK BASICs.

We discovered that the original READY message, at power-up, was in fact, the MEMORY SIZE ? message heavily disguised. This was almost missed in the instruction manuals - about which we will also say more later. We also discovered that to execute a right TAB you must press, simultaneously, the (SHIFT), (CNTRL) AND (I) keys!

Because this unit is already on sale in other parts of the world as the "Video Genie", and probably under other names also, there are no on-screen copyright messages. These have been zeroed out in ROM. Nevertheless, it is the full

Dartmouth/Microsoft Basic, and a small sticker beneath the unit proclaims that the BASIC interpreter is (C) 198Ø by MICROSOFT.

Because of certain hardware changes there are some other small changes in the ROM. These relate to the line-printer operation (on the System 8Ø, the line printer is port addressed only) and to the operation of port 255 (used on the TRS-8Ø for the more sensible 32/64 graphics as well as for cassette output on both units). There is also a change to the READ logic. We have not yet been able to determine the long-term consequences of this alteration, but it is almost certainly of little consequence.

** ** However, the important news is that all the TRS-8Ø ROM routines ARE in the System 8Ø, and ARE located at the same addresses.

** ** The BASIC in the System 8Ø IS TANDY Level II Basic.

** ** There is, therefore, no reason why any TRS-8Ø software, either in BASIC or in machine/language should not work PROVIDED that it does not require any arrow symbols in its video display and also provided that the user will not be required to input either a (CLEAR) or a TAB from the keyboard, and provided that the use of large graphics is not absolutely essential.

In some circumstances the (SHIFT), (CNTRL), (I) method of generating a TAB will not work. Although the function creates an identical result when executing a $tring input, it does not generate the same responses in the keyboard memory, and some of our authors use direct peeking into this memory for speed.

The System 8Ø boasts an inbuilt cassette recorder. We used to think that this was a good idea. Now that we own a System 8Ø, we KNOW that it's a BLOODY SILLY IDEA - unless it works perfectly - and the one on our unit (and according to reports from readers and others all System 8Ø's are the same in this regard) is totally R.S.

Because there is no loudspeaker, or other audio-outlet (not even a socket for an earpiece) it gets amazingly frustrating when the System 8Ø refuses to load a perfectly good tape, since you cannot find out where you are without getting another tape-recorder anyway, which seems to us rather self-defeating!!!!

Because loading the System 8Ø is even more fussy and inconsistent than an unmodified Tandy TRS-8Ø (yes, even such a terrible thing is possible - and Dick Smith has done it!) - thank-you Fort Worth for the XRX III! - you are forced to use the second cassette port. Fortunately the plug is again pin-compatible with the TRS-8Ø! Unfortunately, SYSTEM will not work on #-2, so if you are trying to load a machine language tape you will resort to ripping the System 8Ø apart - to see what gives - and you'll find the most horrible tape recorder East of Aden. In fact, we would say that if this unit were manufactured in Japan (where they give export a capital E) they probably wouldn't allow the unit to leave the country!

Both the mechanism and the electronics of the cassette are woeful. The major reason for our unit failing to load is a grotty little switch, vaguely connected to the record button. This is supposed to switch between play and record. It might succeed if the connection didn't flap about like a barn door in a gale - and since the connection is made of spring steel, it's kind of hard to bend it without breaking it.

The erase/bias oscillator is a radio-technician's idea of purgatory, visibly degrading the otherwise excellent (superior to Tandy) video output whenever you go into record mode. (Transmits well on F.M. too!) If you use the UHF modulated output, you might as well turn the T.V. off while you record, because you'll only get upset, otherwise. (Has the F.C.C. seen this yet?)

There appears to be some sort of constant-volume device, but it certainly doesn't make for easy loading.

The rest of the construction seems to be a strange mixture of good ideas and conservative design, kind of screwed up with odd bits of penny-pinching.

For example, the Keyboard assembly uses a high-quality keyswitch, of the type used in office adding-machines, and is held down by no fewer than EIGHT screws. Yet, the associated metal-work has provision for nine extra keys (and they left two out?) and the pcb. would easily accommodate these. Then there just isn't any physical support for the centre of the keyboard assembly, which will be very necessary when the cassette deck almost begs you to smash your fist through it.

The interconnection between the boards (keyboard and two main p.c.b's) does not use flexible printed-circuit, but steel (!!!) wires in a very rigid clear vinyl former - soldered at one end and plugged in to what looks like a pcb. edge-connector at the other. Strong it is. Electronically, we suspect that it might create some high resistances. Physically - well, it won't break easily, but be prepared to spend two hours re-connecting the socket end if it comes unplugged.

The power-supply is in its own self-contained module. This may be a convenience of manufacture, since the System 8Ø is destined for world-wide distribution. The module has a mesh grid in just the right place to stop little fingers from poking where they might get hurt. At first we thought that this may have been screening to prevent some of the spurious R.F. interference from getting into the mains, but since the screen isn't earthed, and isn't continuous, it seems more likely that it was an afterthought to stop people like us from looking at the power-supply board - it is not the most beautiful piece of wiring that we've ever seen - but it does work.

One nit-picking comment here. The screws that hold the power-supply cover-grid down look exactly like those that hold thekeyboard down. Unfortunately, if you put a keyboard screw into a power-supply screw-hole, you won't notice, until you find that one of the keyboard screws won't do up tightly. By then it is too late, because the power-supply screw-hole is now too big for the right screw. Why not standardise?

The System 8Ø purchased by Micro-8Ø was a 16K unit. Only one I.C. was in a socket, and that was the Z8Ø itself. We didn't like this, because it makes servicing harder than it need be. We sincerely hope that the 4K versions DO use sockets. Dick Smith and other retailers sell memory upgrade kits at very sensible prices, but the ability of your typical 4116 to withstand the average computer hobbiest's soldering is extremely limited and single prime 4116's currently cost around the $14 mark.

The rest of the internal construction was excellent. Pressure of time has precluded our attempting to analyse any of the circuitry, but the tracks on the two main pcb's were well etched and appeared solid, as did all of the joints. Interconnection between the inbuilt cassette and the main unit was through a small but solid plug, as were the connections from the power-supply module. There was no evidence of any dry joints or solder-balls, and with the exceptions noted, the System 8Ø appeared to be a solidly constructed piece of equipment.

Now to the instruction manuals. We thought it surprising that something so very long in the production pipeline as the System 8Ø would have any need for addendum sheets in its manuals, but addendums there are - and some of them very basic ones at that - like how to protect your high memory... how to access machine-language subroutines, etc.

On the whole the manuals are adequate, but we think that most owners will quickly be heading off to their local Tandy Store for copies of the TRS-8Ø literature, to obtain adequate expansions on many of the commands. There is evidence of an attempt on the part of the manual's authors to emulate David Liens witty and pithy style (a la Level I handbook), but it doesn't come off. At least one reader has been trying to get technical literature and mud-maps from Dick Smith Electronics - so far, we understand, without any great success, although there is some hint of "eventually..."

Which brings us to that very question: How long is eventually? Garry Crapp, Service Manager with Dick Smith Electronics, was in Hong Kong during the first week of June, talking to the production engineers at the plant of origin. We understand that he had a whole bag of TRS-8Ø software that just wouldn't load/run on the System 8Ø, and was trying to (a) solve the cassette problems, and (b) get some of the "improvements" removed (like getting those two missing keys put back). The end results are not known to us at Micro-8Ø, but this does show that Dick Smith Electronics have their hearts in the right place.

The System 8Ø was announced as having S1ØØ compatability. It doesn't have any compatability at present. The one outlet port is System 8Ø 5Ø pin bus, presently unique.

Until we heard of Garry Crapp's visit to H.K. we were very doubtful of D.S.E's long-term plans for the System 8Ø. In fact, we had gained the impression that they had lost all interest in the project. We knew of some 5ØØ-7ØØ units coming into Australia. We knew of a sign in some Dick Smith Electronics stores implying that the wait for any upward expansion of the System 8Ø would be indefinite. As late as ten days prior to the Sydney Home Computing Show we were advised by a spokesman for Dick Smith Electronics that they would not be exhibiting the System 8Ø, although, in the event they did. (Perhaps because their stand was next to TANDY?) We heard of an internal Dick Smith Electronics memo asking their Store Managers whether they would be able to sell an additional 5ØØ units Australia-wide if an order were placed. (5ØØ units! Surely if this computer were even 75% of what it was announced to be, they would be talking of 5 or 6 times that figure?) Just mention "upward expansion" to a Dick Smith Electronics store assistant, and you'll be very suddenly told that the "real home computer, of course, is the Exidy Sorcerer".

However, one reader has supplied us with a stack of photocopies - purporting to be circuits of the S1ØØ expansion for the System 8Ø. In fact, the accompanying notes even gave the unit a code number, but the circuits appeared to be of individual S1ØØ boards from within. Nevertheless, one Dick Smith Electronics source has suggested that the wait may only be a couple of months. Certainly, if Dick Smith Electronics don't come out with an expansion, it's London to a Brick that someone else will.
Remember, this computer is coming from Hong Kong (allegedly from a Toy Factory!) and is clearly designed for massive quantity production. Within two years, we predict that the System 8Ø (or Video Genie, or Alpha One, or what-have-you) will be all over the world, under a dozen brand-names, and you will probably be seeing the C.B. boom (and crash?) all over again with a new product. There will be a dozen competing expansions available - EVENTUALLY.

We have been advised of an impending price rise on the System 8Ø, which will bring the 16K version up from $699 to $750 on August 1st.

Now let us do some sums...	System 8Ø with 16K	$ 699.00
	D.S.E. Monitor	149.00
	Total	$ 848.00
	T.R.S. - 8Ø 16K complete with monitor	$1169.00
less	10% discount available to Micro-8Ø subscribers	116.90
	Total	$1053.10

Now, even that sum puts the Dick Smith System 8Ø in a very very favourable light. Even if you buy a decent tape deck from your local Big W and put it in the System 8Ø, you end up at least $150.00 ahead - and that pays for your next 16K of RAM chips.

To sum up then, the System 8Ø is not without its faults. Dick Smith Electronics appears to be trying to fix these problems... The machine has an identical computing capacity to the TRS-8Ø, for a lot less dollars. For any hobbiest, who is not looking for upward expansion in the immediate future,or for someone who wants to make his fortune writing software for the System 8Ø/Alpha One/what-have-you, a bargain! If you want immediate reliability and expandability, we can only say that you were told.

** ** Some additional notes to add to your confusion.

The May 1979 issue of Amateur Radio Action contained a full page advertisement for two "new" Z8Ø based computers, both offering Tandy Level II compatible BASIC, Tandy Graphics, etc. These were the Alpha One and the Alpha Two. The difference between the two models? Alpha Two had full colour capability. The company responsible for that advertisement, we are told, is no longer in existence, although one of the principals is still in the Computer Retail Trade. When I spoke to him, in May and June 1979, I was given to understand that the original intention was to manufacture the Alpha One in Australia, and to import the Alpha Two from Hong Kong. The viability of this scheme depended on obtaining sufficient expressed interest from the various Education Departments throughout Australia. When this was not forthcoming, negotiations with the Hong Kong source were taken over by Dick Smith Electronics, and the original partnership advertising the two units was dissolved. I was told AT THAT TIME that the only problem with the Alpha One was THE CASSETTE DECK!!!
The Dick Smith System 8Ø IS the Alpha One.

It is interesting to speculate "How eventually is the Alpha Two. Is Dick Smith planning a colour version?"

-ooooo-

***** TRS-80 USERS GROUPS *****

MELBOURNE-EASTERN SUBURBS TRS-80 USERS GROUP

1st Wednesday of the month	Tandy Store, 96 Koornang Road, Carnegie
3rd Wednesday of the month	Kingswood College 355 Station Street Box Hill.
Contact:	John Fletcher, 89 0677 between 9 - 4.

ADELAIDE:

1st Thursday of the month	Tandy City Store 94 Gawler Place Adelaide.
Contact:	Rod Stevens, 51 5241 between 9 - 4.

DARWIN: Contact Toni Domigan, P.O. Box 39086, Winnellie, N.T. 5789.

** PROBLEM CORNER ** conducted by the MICRO-80 resident hackers.

Q. "Could you please explain why I can't increment the "step number" in a FOR/NEXT loop (Level I)? I have only got a rough idea of the inner workings of the TRS-80.

```
1Ø C=6: Y=1Ø
2Ø FOR X=1TO124STEPC
3Ø SET (X,Y)
4Ø C=C+1
5Ø NEXTX
```

This does not work as the step remains at 6. I originally tried something similar to this to set vertical lines at incrementing spacing. As C is a variable I thought it could be incremented, but alas - NO GO!"

A. Well now, I cannot agree with you that this doesn't work! It does work in both Level I & II, and it does exactly what the program asks it to do! The problem is that you've not got the problem set out correctly.

If you add a new line ...

```
45 PRINT AT 512,C
```

you'll see that C IS being incremented. The problem that you are experiencing is caused by the fact that the parameters for the loop have already been set, and when they were set the value for C was 6.

To get the results you need, try

```
1Ø C=6:Y=1Ø
20 X=X+C:IFX>124GOTO6Ø
3Ø SET(X,Y)
4Ø C=C+1
5Ø GOTO2Ø
6Ø END
```

which, incidentally, will probably work faster than a FOR/NEXT loop anyway.

Q. "I have a number of Level I programs that I would like to run on my Level II. Is there a program that will convert my Level II to Level I? I have the Tandy CONVersion program, but many of the converted programs do not run and come up with errors that I cannot understand.."

A. There are several "Level I in Level II" programs around. Obviously, we at MICRO-80 need to use most of them from time to time. With Disk systems we use NEWDOS as the operating system, and this package includes not only a Level I in Level II, but the added facility of being able to save Level I programs to Disk.

Unfortunately, there is not any provision for making a hard-copy listing with this version of Level I. We also use LEVELIP - a tape based Level I, which has a LLIST facility, but no provision for saving to Disk. Peter Hartley has both the Level I and Level II ROMS on his C.P.U. and yet another switch added to the growing array on his keyboard. (See article in this edition - Ed.) Probably the simplest solution to your problems, though, would be to resolve the reasons that the converted programs are refusing to run, as the very powerful editing facilities of Level II will quickly allow you to debug the "converted" program.

First of all, the CONVersion program has some bugs in it. These are:-

(a) After conversion, all PRINT AT statements need to have the trailing semicolon changed to a comma

(b) The conversion program will change S.T (STEP T) into STOP so change it back

(c) Any IF POINT statements need a minus inserting before any numeric ones in the statement, if they are to work in Level II

(d) If when the program is RUN, you get a FN ERROR in any SET or RESET or IF POINT statements, you should hit Q (to escape from AUTO EDIT) and then LIST the offending line. Then check the values being SET, RESET or tested. If any of these are outside the Level II graphics parameters (Ø to 127 horizontal; Ø to 47 vertical) you will have to insert a couple of statements to lose the wrap-around graphics values. A subroutine like this:-

```
6ØØØØ IFX>127 X=X-128:GOTO6ØØØØ
6ØØ1Ø IFY>47 Y=Y-48:GOTO6ØØ1Ø
6ØØ1Ø RETURN
```

will generally cure all these problems, but you may also need to make some adjustments for negative values.

- ooooo -

***** MICRO-80's EDUCATION CORRESPONDENT *****

We are sure that there is a large field of application for microcomputers in education both as a teaching aid and for administration. The problem until now ha been that none of the editorial staff are trained educators and we have been wary about publishing programs to be used as teaching aids because of the enormous potential they have for doing harm, if not properly designed.

Recently however, a reader submitted such a program to us for publication which was very well written indeed. We immediately contacted the author and, after a little discussion, he agreed to become our education correspondent.

We welcome Spencer George to the MICRO-80 Team. Spencer is a teacher in Melbourne and you will find the first of an occasional series of his articles in the followi pages. The program which so impressed us, "SYMBOL/BAS" will be published in next month's issue.

- ooooo -

TEACHING WITH A COMPUTER

Spencer George
14/47 Yerrin Street,
BALWYN, 3103.
(03 836 4225)

There are brilliant teachers, and very dull class minders. There are interesting books and boring manuals. Did you enjoy reading, using and trying out the suggestions in the user's manual for Level 1? Most people do.

Did the level II reference manual give you the same reactions ? Most people find it tedious and without a complete index it is not even useful for finding information quickly.

The Level I book teaches, but the Level II manual merely gives information. If you have not yet encountered learning Level II by David A. Lien I suggest that you make sure that you obtain the use of a copy for at least a couple of hours. It is a book which teaches.

Some computer tapes teach because they are interesting, while other tapes give information, but may be boring. Level II basic instruction course lessons are as boring as the Level II manual and you cannot turn back the pages, but you can turn the computer off. However I have seen a number of times groups of men, women and children pushing for their turn to play with the tape containing android NIM because this program challenges the player and makes good use of the graphics capabilities of the TRS-80.

```
**************************************************************************
**************************************************************************
**                                                                      **
**   RULE 1          A program must interest the user and               **
**                   make him work                                      **
**                                                                      **
**                                                                      **
**************************************************************************
**************************************************************************
```

What do you want to teach in your lesson? You must have the ideas clearly in your mind, or better still, on pieces of paper. Some suggestions. Do not try too much at once. Give a small amount of information and then get the student to use the information.

```
##################################################################
##################################################################
##                                                              ##
##  INFORMATION                                                 ##
##  """"""""""""""                                              ##
##                                                              ##
##  Copper, gold, iron, nickel, and uranium are metallic        ##
##  elements.                                                   ##
##                                                              ##
##  Sulfur, nitrogen, carbon, neon and bromine are nonmetallic  ##
##  elements.                                                   ##
##                                                              ##
##                                                              ##
##################################################################
##################################################################
```

```
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
:                                                                    :
:   USING THE INFORMATION.                                           :
:   ......................                                           :
:                                                                    :
:       Is gold a metallic element ?                                 :
:       Is uranium a metallic element ?                              :
:       Is carbon a metallic element ?                               :
:       Are carbon and iron metallic elements ?                      :
:       Are nickel and gold metallic elements ?                      :
:       Are neon and bromine metallic elements ?                     :
:       Are copper and uranium metallic elements ?                   :
:                                                                    :
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
```

```
*****************************************************************
*****************************************************************
**                                                             **
**  USING THE INFORMATION.                                     **
**  ......................                                     **
**                                                             **
**    Will gold conduct electricity ?                          **
**    Will sulfur conduct electricity ?                        **
**    Will carbon conduct electricity ?                        **
**    Will a mixture of copper and nickel conduct electricity ? **
**    Will a mixture of nitrogen and neon conduct electricity ? **
**                                                             **
**                                                             **
*****************************************************************
*****************************************************************
```

You might teach two major ideas and perhaps twenty variationsfrom these major ideas in one lesson.

Now that you have written what you want to teach, consider what you will require the student to already know before he starts the lesson. The example assumes that the student is familiar with the words metallic and element. He must have acquired that information elsewhere. (Perhaps a previous lesson)

If a lesson is suitable for computer teaching it must be presented as a series of 64 x 16 characters on a screen. On sheets of paper display what you want to appear on the video screen. Remember many ideas are not suitable for computer teaching. Be imaginative, use the graphics. Then try out your paper suggestions on some volunteer. Make a written note of all items which to the guinea pig seem obscure, or are interpreted differently to your intention, and of course, note your own mistakes. Correct your paper displays.

Now for the programming. I find there are three principal key phrases for successful teaching on a computer:

```
CLS
PRINT@
INPUT
```

Use CLS to keep the screen uncluttered. Remove information which is no longer necessary. Keep each display simple.

PRINT@   ;   allows you to place information on the screen in any section you wish without altering information on any other section of the screen. The semicolon (;) at the end of the line stops the automatic linefeed which produces undesired effects when the normal print statement is used.

PRINT@   ;   also will overwrite a section of the display. Try

```
PRINT@340, "TEACHING IS FUN";
```

Then try

```
10 FOR J = 0 TO 1023
20 PRINT@ J, CHR$(191);
30 NEXT J
40 PRINT@ 340, "TEACHING IS FUN";
RUN
```

A student must react to a program. Input statements allow questions to be asked and the answers to be checked by the computer, but writing program segments which check answers is not always easy.

```
   45 INPUT "Will gold conduct electricity"; Z$
   50 If Z$ = "YES" then GOSUB 100
                         else PRINT "Gold is a metal and therefore conducts
electricity."
   55 INPUT "Will sulfur conduct electricity";  Z$
      .
      .
      .
```

```
100 FOR J9 = 1 TO 12
110 PRINT TAB(J9); "RIGHT"
120 NEXT J9
130 RETURN
```

An input statement temporarily halts the basic program, so it may be used to provide time for an individual to read all the information on the screen and then when the individual is ready give the next set of information. A suitable subroutine is

```
    2440 FOR JH = 1 TO 25
    2450 X$ = INKEY$
    2460 IF X$ <>     ""    RETURN
    2470 PRINT @ 970, "PLEASE TOUCH THE SPACE BAR WHEN READY";
    2480 NEXT JH
    2490 FOR JH = 1 TO 25
    2500 X$ = INKEY$
    2510 IF X$ <>     ""    RETURN
    2520 PRINT@ 970, "
    2530 NEXT JH
    2540 GOTO 2440
```

Tables in which information is stored may be used to produce a set of questions without typing out each individual item each time it is used in a program. Random and RND( ) in the program choose a selection from the list.

```
    209 DIM Y$(15), Y3$(15)
    211 FOR J = 1 TO 10
    213 READ Y$(J)
    215 NEXT J
    217 DATA COPPER, POTASSIUM, ZINC, NICKEL, IRON, ALUMINIUM, URANIUM, GOLD,
CARBON, NEON, BROMINE, SULFUR, HELIUM, OXYGEN, NITROGEN
    219 '
```

This sets up the set of metals and nonmetals

```
    221 FOR J = 1 TO 8
    223 Y3$(J) = "Y"
    225 NEXT J
    227'
```

This places a Y for yes in the first 8 rows of Y3$( )

```
    229 FOR J = 9 TO 15
    231 Y3$(J)  =  "N"
    233 NEXT J
    235 '
```

This places a N for no in the next 7 rows of Y3$( )

```
237 FOR J = 1 TO 10
239 L = RND(J)
241 CLS
243 PRINT @ 128, "IS";  Y$(J);  " A METAL ";
245 INPUT Z$

.

.

.

.
257 NEXT J
259 '
```

This produces ten random questions of the form is ..... a metal ?

Make sure that loops for sets of questions are not too long or students will become bored.

Information may also be printed on a lineprinter. When an error is made it is useful if a sentence with the correct information is displayed on the screen. A permanent record is produced on the lineprinter and the program repeats the original question

In our example the student's response is in Z$. The expected response is either yes or no, so the first letter of a correct response will be either Y or N. We have already stored the correct responses in Y3$( ).

```
  247 IF LEFT$( Z$,1 ) = Y3$(J)  THEN  CLS;
                                      FOR J3 = 1 TO 12;
                                      PRINT " CORRECT";
                                      NEXT J3
                                 ELSE GOSUB 2000

 2000 IF Y3$(J) = "Y"        THEN W$ = "METAL."
                             ELSE W$ = "NONMETAL."
 2010 PRINT   Y$(J);     " IS A ";   W$
 2020 LPRINT Y$(J);      " IS A ";   W$
 2040 RETURN
```

It is a good idea to produce programs which work whether a lineprinter is connected or not. This is done by ignoring LPRINT statements when the printer is not ready or attached. To achieve this, statements should take the form

```
 IF PEEK(14312)  =  63 THEN LPRINT    ......
```

You will notice that the student is rewarded when the expected response is given, while the correct information is displayed when there is an unexpected response.

```
***************************************************************
***************************************************************
**                                                           **
**                                                           **
**      RULE 2          Reward correct responses             **
**                                                           **
***************************************************************
***************************************************************
```

Well you have written a program. Now try it out yourself a number of times. Deliberately make errors and see what the program does. Eliminate all the bugs which you find.

Then try your program on your friends, son, mother and students. After you have tried it on about ten people you will have found most of its bugs. A word of warning. If you are a teacher and plan to use the program with your own students then use only the debugged final program on them. Use a different set of people to help you find the errors. Students (in fact all people) quickly lose confidence in programs which frequently do not operate correctly, so remove the bugs before giving the program to its critical audience.

```
***************************************************************
***************************************************************
**                                                           **
**                                                           **
**       RULE 3       Programs must work correctly for       **
**                    the normal user.                       **
**                                                           **
***************************************************************
***************************************************************
```

Now that the program is working as you planned, add plenty of comments, so that you will know what each section is doing. Add sufficient so that you will still be able to understand it when you want to improve it in three year's time. Also if the program is good, you will want to give (or sell) it to others. Appropriat comments will allow them to adjust your program to their particular teaching needs. Remember that the single quote mark (') is a quick and tidy symbol for REM in program listings. Use the down arrow to give spaces and new lines.

```
        570 '       THIS IS MY FIRST REMARK
                    THIS IS A VERY LONG COMMENT WHICH OCCUPIES A NUMBER OF
                    LINES WITH USEFUL INFORMATION.

                    I HOPE YOU GET THE IDEA
        580
```

While documenting programs make sure that all operating instructions are shown when needed on the video screen. We all tend to lose that vital piece of paper giving the special set of instructions. Any person should be able to operate the program by typing RUN and then following the screen directions.

This means if you want a student to type his christian name you do not use

```
          INPUT Z$

                    WHEN ONLY

                                        ?
```

Will appear on the screen, but you program

```
          CLS
          INPUT "PLEASE TYPE YOUR CHRISTIAN NAME
                AND THEN TOUCH THE ENTER KEY";     Z$
```

so that

```
          PLEASE TYPE YOUR CHRISTIAN NAME
          AND THEN TOUCH THE ENTER KEY      ?
```

will appear on the screen.

This article contains a number of suggestions. Please try them out with your own programs.

***** INPUT/OUTPUT *****

Letters to the Editor

From: Richard B. Kiltie, West Coast Community College, Queenstown, Tas.

You asked for INPUT on your Assemble Language articles. Before I begin, let me say that I'm "into micros" because my College bought an APPLE II and a TRS-80 (I favoured 2 80s) to alleviate difficulties we were having with our Ed. Dept. time-sharing TASNET. So, I had to turn from DEC equipment to learn about micros.

1. Absence of diagrams in your articles.

2. Your "beginning" articles mentioned Editor/Assembler, Monitor, Disassembler, without definition or background. Since your readers were presumably ignorami, wasn't this premature, eh??

3. Units - have they changed? Used to be 1000 bits = 1 K (really 1024 bits = 1 K); now it's 1000 bytes = 1 K (really 1024 bytes = 1 K) ?? Is 1 byte = 8 bits ??

4. In your first article, you assert that a register is merely a memory cell; whereas Arnold <u>et al</u> (Mod. D.P.) assert that a register is a device to receive, hold, and transfer data according to the control unit. There is an important hardware difference here: a memory cell is passive, one can write into or read from it. Whereas a <u>device</u> involves more than this.

5. Are the blank spaces in your M/L, A/L the way to type in a program?

(Thank you for your comments, Richard. We certainly do like feedback. To take your points in order:

1. You're right, the early articles, in particular, would have benefitted from a few more diagrams (we did have one!!) Actually, that would apply as a general comment on all our early articles. As we get older and more experienced we are including more diagrams so I think we have learned this lesson.
2. I don't like the word ignorami but your point is made. The next issue will carry a definition and brief description of each of these utilities (We will even explain what a utility is!)
3. Units - ah what a murky subject. First of all, the safe ground. K in computer science is the number 1024 (= 2^{10}). You can have 1 K bits of data or 1 K of CPU's! But what is a byte? In the beginning there were bits and data words. A word was the number of bits which a computer stored, moved etc. at one time. Hence, one computer could use a word length of 16 bits and another a word length of 32 bits. As far as we are able to ascertain, the term byte was synonymous with word length, so that 1 byte of data in the former computer would consist of 16 bits - but in the latter, it would consist of 32 bits. Then, along came the 8 bit microprocessors and, more importantly, hundreds of thousands of computer owners who had never had anything to do with main frame computers or even minis (we include ourselves in that merry band). The term "byte" was adopted to mean 8 bits. So now by sheer force of numbers that's what it is and you will see more erudite publications than ours (yes, there are some) which will refer to the "upper and lower bytes" of data in a 16 bit microprocessor system, for example. So a byte is 8 bits, certainly as far as MICRO-ten * byte is concerned!
4. Arnold et al might well be right when talking about main-frame computers, minis, etc. - maybe a register is more than a memory cell, although if you have quoted Modern Data Processings' exact words, they do seem rather ambiguous. In hardware terms, a register in the Z80 or any other microprocessor we know of, is a set of flip-flops (usually 8 or 16). And that is exactly what a static RAM cell is, whilst a dynamic RAM cell of the type used in the TRS-80 behaves in a similar fashion. It is true that the registers, particularly the A register, or accumulator, are intimately bound up with other parts of the microprocessor such as the ALU (arithmetic logic unit) but memory cells they are, no more and no less. Just to prove that we have read a book, William Barden Jnr. in his book "How to Buy and Use Minicomputers and Microcomputers", defines a register as "a high speed set of flip-flop storage devices".
5. The blank spaces in an assembly language listing are the way the program is typed into an editor/assembler program which subsequently produces the actual object code which is the machine language program. We will give a fuller explanation when we describe the Editor/assembler, monitor and disassembler programs next issue. In the meantime, I hope this has cleaned up the main points and, once again, thank you for your feedback. - Ed.)

***** HARDWARE SECTION *****

LEVEL II + I BASIC - by Charlie Bartlett
INCLUDING - 4K TO 16K CONVERSION

This article explains how to make the following modifications to your TRS-80;

1) Convert your Level I or Level II machine from 4K to 16K of RAM

2) Upgrade your Level I machine to Level II

3) Have both Level I and Level II in the same machine.

Simple, step-by-step instructions are given for each of these conversions. The memory upgrade, in particular, needs no soldering iron nor any particular electronic knowledge. Both the Level II upgrade and Level II + I conversion do, however, require the use of a soldering iron and should only be carried out by someone who has had previous experience of working with MOS integrated circuits. A low power (20 - 30 watt) soldering iron with an EARTHED TIP is required for these modifications. MICRO-80 and its staff have carried out a number of conversions using the procedures described here and we are satisfied that they work. However, we can take no responsibility for any loss or damage which might occur when they are carried out by our readers.
WARNING: THE MODIFICATIONS DESCRIBED IN THIS ARTICLE WILL VOID YOUR TANDY WARRANTY.

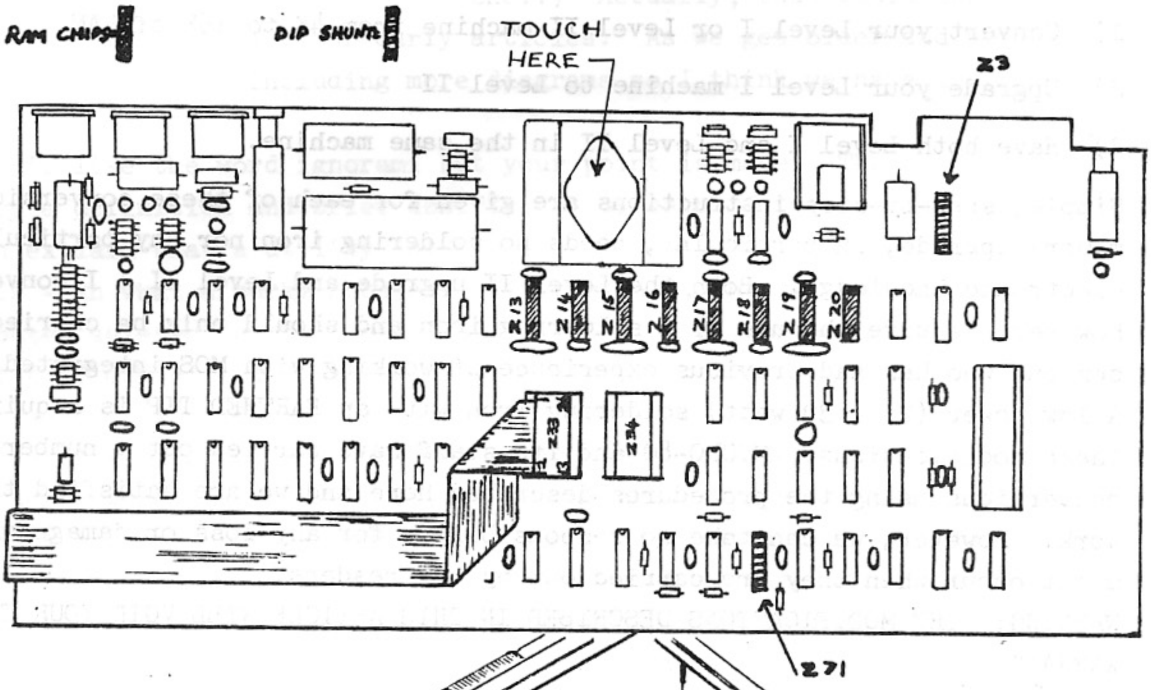
INSTALLATION OF 16K RAM

The installation of the 16K memory chips is simple, all you need is a small flat bladed screwdriver and a Philips screwdriver. Here we go;

1. Turn the computer upside down (on something soft so you don't scratch it) and remove the six Philips screws.
2. Turn the computer keyboard right side up and remove the top cover.
3. Carefully lift the keyboard from its five supports and let it hinge towards you. While holding the keyboard in one hand remove the five white plastic spacers, noting where they were inserted (refer Fig. 2).
4. Then remove the entire printed circuit board from the case and lay it down, component side up, on a table with the keyboard above it.
5. The 4K memory chips now need to be removed from the board. These chips are numbered Z13 through Z20, refer Fig. 1 for their exact location. Observe which way round the chips are installed. There should be a small notch, or dot or 1 at one end of each chip. This should point towards the top of the board in Fig. 1. Remember this orientation as you will need to instal the 16K chips the same way round.

NOTE: To equalise any possible static electric change, JUST BEFORE you unwrap or handle the IC's, take the package in your hand and touch the large metal cased transistor (see Fig.1) with your finger. Repeat before you pick up each IC or remove one.

FIGURE 1



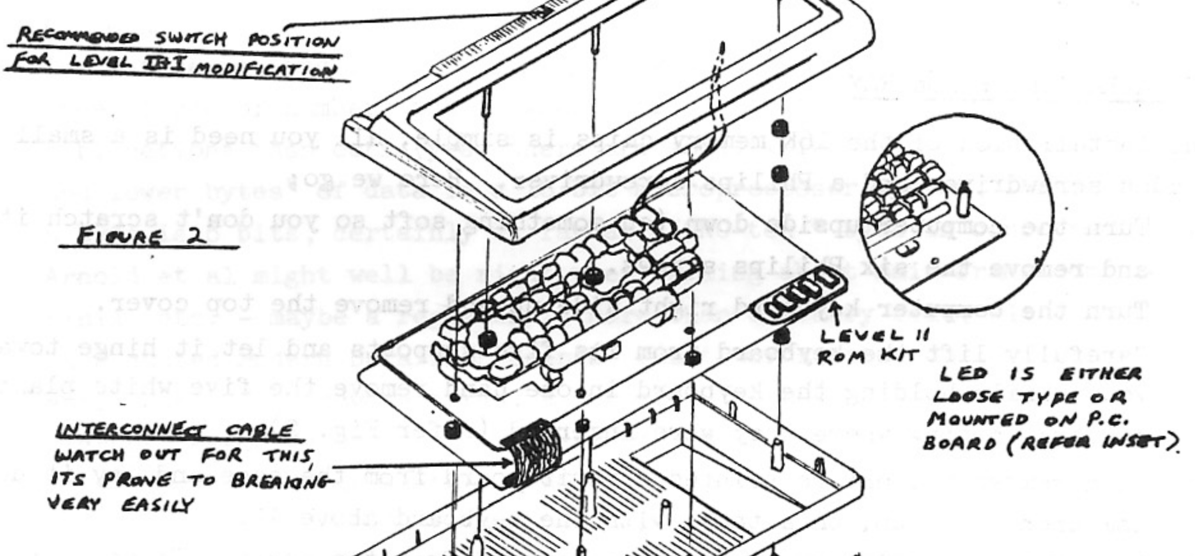

RECOMMENDED SWITCH POSITION FOR LEVEL II-I MODIFICATION
FIGURE 2
LEVEL II ROM KIT
LED IS EITHER LOOSE TYPE OR MOUNTED ON P.C. BOARD (REFER INSET)
INTERCONNECT CABLE WATCH OUT FOR THIS, ITS PRONE TO BREAKING VERY EASILY
1 3/4"
2"
1"
1 3/4"

FIGURE 2

To remove one of these chips, slip the flat blade of the screwdriver under one end of the chip and pry it up slightly, then, pry the other end up slightly until the chip can be removed. Remove the other seven chips in the same manner. Be careful as the pins can be bent very easily and you may want to save the 4K chips to use in a board design of your own. Put them in the container that the new chips came in as it is conductive and will protect them.

6. Instal the new 16K chips in the empty sockets. Be sure to put them in the right way up, a small notch or dot or 2 should appear at one end of the chip, place this towards the top of the board, Fig. 1.
7. Remove the two dip shunts Z-3 and Z-71 using the same removal technique described before. The position of these is shown in Fig. 1. Instal the new dip shunts and program them according to the chart in Fig. 4. On machines with an 'A' board, the 16 pin dip shunt will not fit into the 14 pin socket. Instal the dip shunt so that pins 8 and 9 hang over the end of the socket. (Those are the pins at the opposite end from the notch).
8. If you are not installing Level II, reassemble your computer, noting that some screws are longer than others. (see Fig. 2). Turn on your computer, type "PRINT MEM." If you have a Level I machine the number should be 15,871. If you have a Level II the number should be 15,572. If the correct number is not displayed recheck the programmable dip shunts for mistakes and try again. If the number is still wrong, the problem could possibly be a bad memory chip.

INSTALLATION OF LEVEL II

The Level II BASIC kit must be ordered from TANDY. The kit contains a preassembled printed circuit board, ribbon cable, resistor, 1 dip shunt, four cassettes and manuals. If you want to make the Level II + I modification described later, you must instal the Level II kit yourself. If TANDY do it, they not only charge you $30 but they have the nasty habit of keeping your Level I BASIC chip, which of course means that any of you that have already had TANDY instal the Level II kit in your computer will not be able to make the Level II + I modification. Besides that, who wants to pay $30 for something as easy as what follows?

NOTE: Level II machines are extremely sensitive to the volume setting on the cassette recorder when CLOADing. TANDY is aware of this and has produced a small PC board called the XRX-III modification. This board should come with your Level II ROM kit. If it doesn't, ask for it. It is available free to all Level II owners. We cannot recommend this modification too highly. It will save you hours of frustration. Next month we will publish instructions for installing the XRX-III.

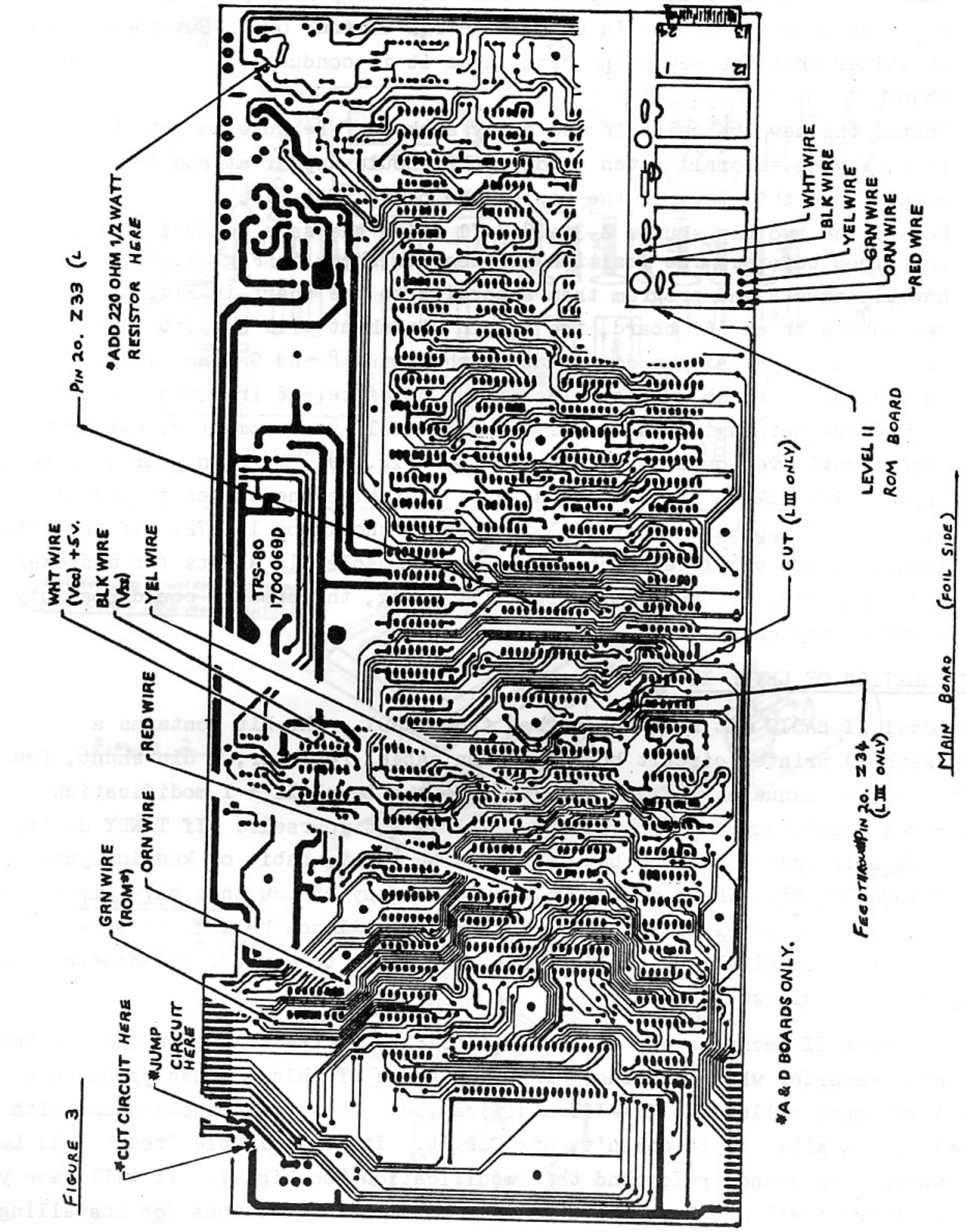


FIGURE 3
PIN 20. Z33 (L
*ADD 220 OHM 1/2 WATT RESISTOR HERE
WHT WIRE (Vcc) +5V.
BLK WIRE (Vss)
YEL WIRE
TRS-80 1700069D
RED WIRE
ORN WIRE
GRN WIRE (ROM*)
*JUMP CIRCUIT HERE
*CUT CIRCUIT HERE
*A & D BOARDS ONLY.
FEEDTHROUGH PIN 20. Z34 (LIII ONLY)
CUT (LIII ONLY)
LEVEL II ROM BOARD
WHT WIRE
BLK WIRE
YEL WIRE
GRN WIRE
ORN WIRE
RED WIRE
MAIN BOARD (FOIL SIDE)

FIGURE 3.

Disassemble your computer as described before, then examine the component side of the board; early model TRS-80's had chips in both Z33 and Z34 sockets (Fig.1) If yours is one of these you will not be able to make the Level II + I modification described later and the installation instructions described for Level II in this article will NOT apply.

However, the majority of TRS-80's have only one ROM chip, either in socket Z33 or Z34 and it should be labelled "BASIC 1". The other socket will be empty.

1. Turn the board over. Before going further, earth yourself as described before, then remove the Level II ROM board from its protective package. Remove the covering from the adhesive strips on the Level II board. Firmly press this board into place as illustrated in Figures 2 and 3. The six coloured wires in a plastic sleeve should extend out to the left of the board, the connection of these will be described later. The 24 pin ribbon cable should be plugged into the Level II ROM board (carefully, so you don't bend the pins) and extended out to the right of the board for the moment. Turn the main board over again, component side up. If your Level I BASIC ROM is in Z33 and you are going to make the Level II + I modification, re-insert it into Z34 (Fig. 1) otherwise remove it altogether. Route the ribbon cable along the board as pictured in Fig. 1 and plug it into Z33.
2. Turn the main board over again. Now to connect those six wires. 3 of them go to the computer address bus, one is the enable line for the Level II ROMs and the other two supply power to the Level II board. Examine Fig. 3 carefully, connect the six coloured wires to the points indicated. (If making the Level II + I modification, leave the green ROM wire disconnected).
3. Connect the resistor as shown (Fig. 3). The purpose of the resistor is to reduce the amplitude of the incoming cassette audio signal.
4. An etch cut should now be made where indicated (Fig. 3) as a safety precaution to prevent more power being drawn through the edge connector. A jumper is made where indicated (Fig. 3) to change this connection to ground. If you are not interested in the Level II + I modification, you are almost finished.
5. Program the dip shunts Z3 and Z71 as shown in Fig. 4.
6. Reassemble the case, turn on the power, when the monitor displays MEMORY SIZE you are on your way to a whole new world with Level II.

LEVEL II + I INSTALLATION

Before we start, you should have the Level II ROM, ribbon cable plugged into Z33, the Level I BASIC ROM plugged into Z34 and the green Level II ROM wire unconnected. A double pole double throw switch is needed to make these modifications.

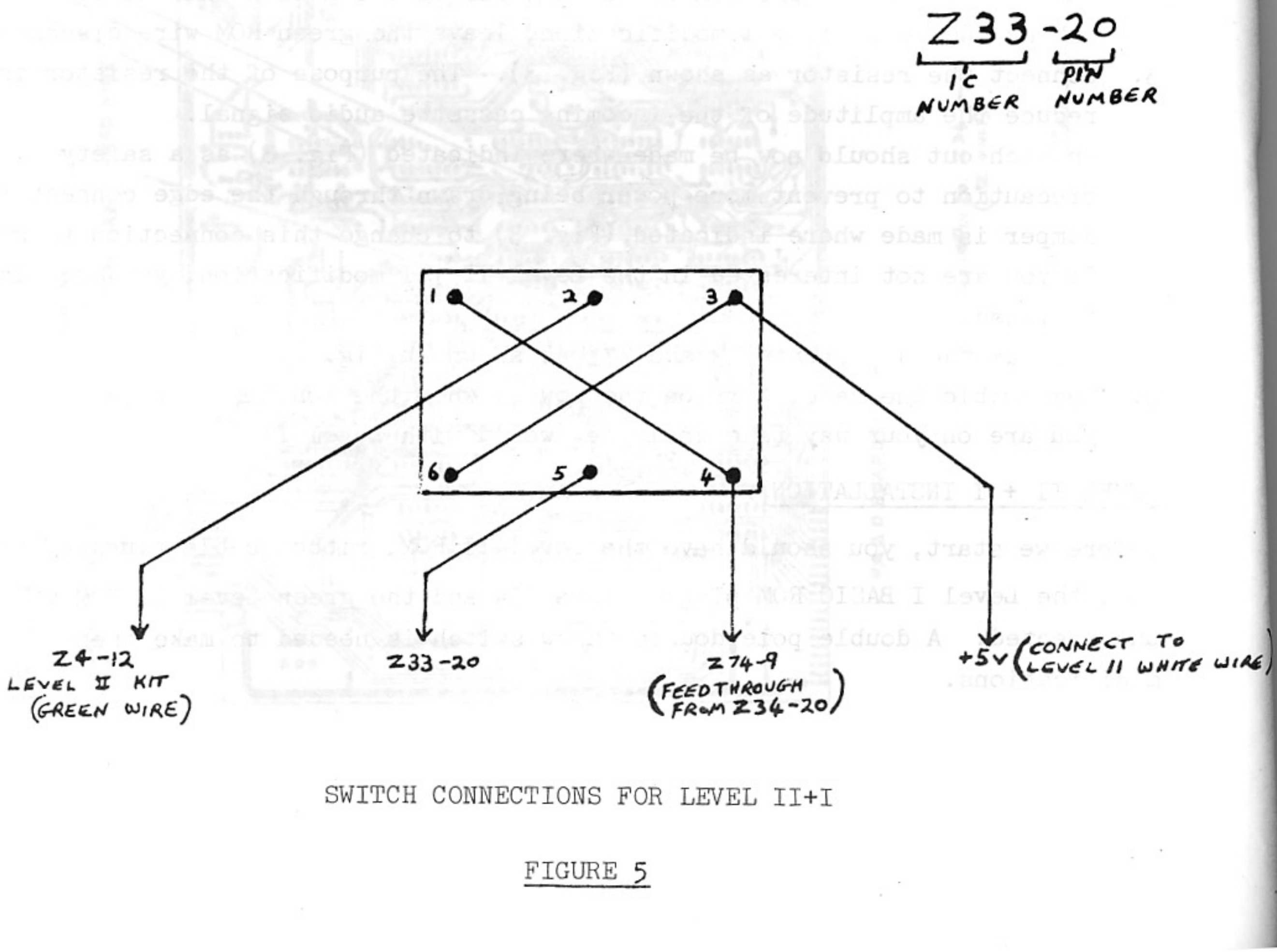
FIGURE 4

DIP SHUNT PROGRAMMING CHART

DIP	SWITCH	4K Level I		16K Level I		4K Level II		16K Level II		4K Level II+I		16K Level II+I	
PINS	number	Z3	Z71	Z3	Z71	Z3	Z71	Z3	Z71	Z3	Z71	Z3	Z71
1,16	1	C	O	C	C	C	O	C	C	C	O	C	C
2,15	2	C	C	C	O	C	C	C	O	C	C	C	O
3,14	3	O	O	C	C	O	O	C	C	O	O	C	C
4,13	4	O	C	C	O	O	C	C	O	O	C	C	O
5,12	5	O	O	C	C	O	O	C	C	O	O	C	C
6,11	6	O	C	O	O	C	C	C	O	C	C	C	O
7,10	7	C	O	C	O	C	O	C	O	C	O	C	O
8,9	8	O	O	O	O	C	O	C	O	C	O	C	O

C = CLOSED

O = OPEN


Z33-20
I'C NUMBER
PIN NUMBER
1
2
3
6
5
4
Z4-12 LEVEL II KIT (GREEN WIRE)
Z33-20
Z74-9 (FEEDTHROUGH FROM Z34-20)
+5V (CONNECT TO LEVEL II WHITE WIRE)

SWITCH CONNECTIONS FOR LEVEL II+I

FIGURE 5

Number 26 insulated wire is recommended for the jumpers, which should be about 12 inches long to allow for proper positioning of the switch.

1. Examine Fig. 5, connect the green ROM wire to pin 2. This is the chip enable for the Level II ROMs.
2. Contact 5 should be made to the back side of the board at pin 20 of Z33. This is the chip enable for the Level I ROM.
3. Contact 3 and 6 should be jumpered together with a short piece of wire and then a jumper from pin 3 is connected to the same location as the Level II ROM board white wire. This is +5V, and holds the chip enable line of the unused ROM in the inactive high state, as the enable lines are active low.
4. The last remaining contacts, 1 and 4 are also jumpered together, then a jumper from these pins is connected to a feedthrough located below pin 20 of Z34 (Fig. 3). This is the chip enable sent from the Z80 microprocessor to the BASIC ROM's.
5. An etch cut must now be made just above the feedthrough that we just connected (Fig.3). If this etch cut is not made, the Level I ROM will never be disabled.
6. Program the dip shunts as shown in Fig. 4 for Level II + I.
7. The last job is to mount the switch in the case.Since switching from Level I to Level II (or vice-versa) with the power on, will cause program errors, you are strongly advised to mount the switch on the sloping back face of the computer where indicated in Fig. 2, to avoid accidental operation of the switch while programming. Not only that, but this position is well clear of the PC board.
8. Put the case back together as described before and power up. With the switch in one position you will have Level I and in the other position, Level II. Welcome to Level II + I. I hope you enjoy all those Level I programs that you thought were gone forever because they were too long to convert. Oh yes, don't forget to change the volume setting on the cassette deck when changing from one level to the other.

NOVICE NOTES

These notes are provided to help readers who are not experienced in working with IC's, but note the warning at the beginning of this article.

1. Dip shunt are programmed by inserting them into their sockets and then using a small flat bladed screwdriver, applying just enough pressure to the metal bars to break them. TANDY's dip shunts are not programmed when you buy them although they have a plastic covering which makes them look as though they are!
2. Don't apply too much heat to the PC board with the soldering iron, you might cause the copper tracks to peel off or damage components due to overheating.

3. Don't handle the components until you have grounded yourself.

4. Wires to be connected should be 'tinned' first. (If you don't know what that means you are strongly advised to seek assistance from somebody before you proceed with modifications or not to proceed at all).

5. The resistor can be soldered in any way round; polarity is not important with this component.

6. IC's must be inserted the correct way round (refer main text).

7. Pay close attention to the pins on IC's when inserting them, as it is easy to bend a pin underneath the IC.

8. The interconnect cable is very fragile, handle the boards gently. If it does break you will have to replace the whole thing with wires.

9. Find a clear area to work in where you wont have to move and not on a nylon carpet PLEASE.

10. Golden rule number ten. If in doubt, 'don't'. That's an expensive computer you've got there; get experienced help.

- 00000 -

** BEETHOVEN'S HELPER (a music chord program)
LEVEL I 4K by P. Ereaut.

This program demonstrates a more serious application for the "baby" '80, and we complement Phil on this unusual application, which shows the correct notes for any one of 140 music chords.

The program is based on the logical assembly of chords from an equally logical sequence of semitones (which the Chineese don't use, which is why oriental music sounds as odd to us as ours must to the Chineese).

When the user selects his requirements, the numeric value of the chord is calculated in lines 70 to 110. Lines 600 to 705 then read from the "Alpha" data, using FOR NEXT loops and the A $tring (which cleverly solves some of the $tring limitations of Level I). The PRINT ATs for the graphics are calculated in lines 415 to 465, and are also read from Data lines, whilst subroutine 410 forces the program to read data from 49 onwards, allowing for the reading of numeric data from mixed alphabetical and numeric data.

Music Note Data (lines 670 to 685) is set up to give the correct notation for chords using sharps and flats. (The musical genius amongst us need not complain about the slight inccuracies for the 5th in the Augmented chords C, G and F, where A& is used in place of G# - it sounds like the same note, even to the most purist of us. If you can scrounge up about 50 bytes, you could add...

```
52 IF((Q=30)*(L=4))+((Q=25)*(L=4))+((Q=32)*(L=4)) Q=Q-24
```

O.K?)

Lines 155 to 340 set the full graphics on the first run, only, and line 155 saves time by bypassing the full reset on subsequent runs.

This will only just squeeze in to 4K (20 bytes spare at run-time) and full use of abbreviations must be made.

```
1 REM * BEETHOVEN'S HELPER - MUSIC CHORD PROGRAM
2 REM * COPYRIGHT (C) 1980 P. EREAUT
3 REM *                       110 PARK ST., MOONEE PONDS, VIC
4 REM * MICRO-80 ISSUE 7, JULY 1980
5 REM LEVEL I 4K
10  CLS
20  P.A.21,"MUSIC CHORD PROGRAM"
21 P.
22 P."PROGRAM WILL PRINT OUT,AND SHOW GRAPHICALLY THE SELECTED CHORD"
23 P."THE CHORD CAN BE MAJOR,MINOR,DIMINISHED,OR AUGMENTED"
24 P."IT CAN BE A TRIAD,OR HAVE AN ADDED 7TH OR 6TH."
30  X=0:H=0
35  P.A.588,"G& D& A& E& B& F  C  G  D  A  E  B  F# C#"
40  P.A.652,"31 26 33 28 35 30 25 32 3 10  5  12  7  2 "
45 IN."ENTER NUMBER OF THE CHORD REQUIRED";Q
50 IN."1= MAJOR, 2= MINOR, 3= DIMINISHED, 4= AUGMENTED";L
55 IF L=3 D=4:G.70
60 IF L=4 D=1:G.70
65 IN."1= NO 7TH, 2= MINOR 7TH, 3= MAJOR 7TH, 4= 6TH";D
70 ON L G.75,80,85,90
75 A=Q+4:B=A+3:G.95
80 A=Q+3:B=A+4:G.95
85 A=Q+3:B=A+3:G.95
90 A=Q+4:B=A+4:G.95
95 ON D G.115,100,105,110
100 E=Q+10:G.115
105 E=Q+11:G.115
110 E=Q+9 :G.115
115 IF X>=1 T.380
150 CLS
155 P.A.23,"GRAPHIC CHORD CHART    (P.E. JAN. 1980)"
160 Y=3
165 F.X=1TO121:S.(X,Y):N.X
170 Y=4:X=1
175 S.(X,Y)
180 F=6:G=11
185 F.Z=FTOG:S.(Z,Y):N.Z
210 F=G+5:G=F+5
215 IF G=31 G=26
220  IF G=66 G=61
225 IF G=91 G=86
230 IF G=126 G=121
235 IF Z>121 T.245
240 G.185
245 Y=Y+1
247 IF H=0 T.250
248 IF Y>14 T.35
250 IF Y>16 T.260
255 G.175
260 X=1:T=0
265 IF X>121 T.300
270 S.(X,Y)
275 T=T+1
280 ON T G.285,290,285,290,290,290,285,285,290,285,290,290,290, 285
285 C=8:G.295
290 C=9
295 X=X+C:G.265
300 Y=Y+1
```

```
305 IF Y=24 T.320
310 G.260
320 F.X=1 TO 121:S.(X,Y):N.X
325 P.A.450,"C";:P.A.455,"D";:P.A.459,"E";:P.A.463,"F";
330 P.A.468,"G";:P.A.472,"A";:P.A.476,"B";:P.A.480,"C";
335 P.A.485,"D";:P.A.489,"E";:P.A.493,"F";:P.A.498,"G";
340 P.A.502,"A";:P.A.506,"B";
380 GOS.900
400 GOS.410
405 G.415
410 F.N=1 TO 48:READ A$:N.N:RET.
415 F.N=1 TO Q:READ K:N.N
420  F.N=1 TO 12:READ W:N.N:RESTORE
425 GOS.410
430 F.N=1 TO A:READ R:N.N
435 F.N=1 TO 12:READ V:N.N:RESTORE
440 GOS.410
445 F.N=1 TO B:READ M:N.N
450 F.N=1 TO 12:READ S:N.N:RESTORE
455 GOS.410
460 F.N=1 TO E:READ J:N.N
465 F.N=1 TO 12:READ U:N.N:RESTORE
470 P.A.K,"1";:P.A.W,"1";
480 P.A.R,"3";:P.A.V,"3";
490 P.A.M,"5";:P.A.S,"5";
500 IF L=3 T.515
505 IF D=1 T.600
510 IF D=4 T.525
515 P.A.J,"7";:P.A.U,"7";:G.600
525 P.A.J,"6";:P.A.U,"6";
600 F.N=1 TO Q:READ A$:N.N:RESTORE
605 ON L G.610,615,620,625
610 P.A.578,"MAJOR CHORD OF ";A$;"  ";:G.630
615 P.A.578,"MINOR CHORD OF ";A$;"  ";:G.630
620 P.A.578," DIM. CHORD OF ";A$;"  ";:G.630
625 P.A.578," AUG. CHORD OF ";A$;"  ";:G.630
630 ON D G.660,635,640,645
635 P."+ MINOR 7TH";:G.660
640 P."+ MAJOR 7TH";:G.660
645 IF L=3 T.655
650 P."+ 6TH";:G.660
655 P."+ DIM. 7TH";
660 F.N=1 TO Q:READ A$:N.N
670  DATA C,C#,D,D#,E,F,F#,G,G#,A,A#,B
675  DATA C,C#,D,D#,E,F,F#,G,G#,A,A#,B
680  DATA C,D&,D,E&,E,F,G&,G,A&,A,B&,B
685  DATA C,D&,D,E&,E,F,G&,G,A&,A,B&,B
690  DATA 386,260,391,265,395,399,273,404,278,408,283,412
692  DATA 416,290,421,295,425,429,303,434,308,438,313,442
695  DATA 386,260,391,265,395,399,273,404,278,408,283,412
697  DATA 416,290,421,295,425,429,303,434,308,438,313,442
700  DATA 386,260,391,265,395,399,273,404,278,408,283,412
705  DATA 416,290,421,295,425,429,303,434,308,438,313,442
720 ON L G.725,730,730,725
725  P.T.(35);A$;"---";:G.735
730 P.T.(35);A$;"--";:G.735
735 RESTORE
740 F.N=1 TO A:READ A$:N.N
```

```
745 ON L G.750,755,750,755
750 P.A$;"--";:G.760
755 P.A$;"---";:G.760
760 RESTORE
765 F.N=1 TO B:READ A$:N.N
770  IF L=3 T.785
775 ON D G.805,780,785,790
780 P.A$;"--";:G.795
785 P.A$;"---";:G.795
790 P.A$;"-";
795 RESTORE
800 F.N=1 TO E:READ A$:N.N
805 P.A$:RESTORE
806 P.A.706,"         TONIC       1ST INV.    2ND INV.     3RD INV."
807 P.A.770,"TRIAD: 1 3 5      3 5 1       5 1 3 "
808 P.A.834,"7/6TH: 1 3 5 7    3 5 7 1     5 7 1 3      7 1 3 5 "
814 P.A.910,"TO SELECT NEXT CHORD PRESS ENTER";:IN.A$
815 P.A.K," ";:P.A.W," ";
820 P.A.R," ";:P.A.V," ";
825 P.A.M," ";:P.A.S," ";
830  P.A.J," ";:P.A.U," ";
835 GOS.900
850 X=1:Y=12:H=1:G.175
900 P.A.576," ":P.A.640," ":P.A.704," ":P.A.768," "
901 P.A.832," ":P.A.896," ":RET.
```

** LUNALERT ** Level I 4K by Luke Brindley

This game simulates the now-traditional "computer-failed-now-it's-up-to-you" lunar landing, but with some interesting variations, using a "radar screen" to show the terrain below - always made up of two mountains and a plateau, but the plateau structure is randomised (lines 1040 to 1120, with the randomiser in 300).

You have no control for the first 10 seconds. Thereafter you enter both horizontal and vertical thrust every 10 seconds (negative values are O.K. - necessary, in fact, i- you don't want to hasten your plunge downward!)

The display gives you :-

VX ... horizontal velocity
SX ... horizontal displacement from point of origin
VY ... vertical velocity
SY ... vertical displacement
AX ... last entered horizontal thrust
AY ... last entered vertical thrust

The point of origin is always the top left corner of the screen, but you'll really only need SX and SY when you've goofed and blasted off the screen limits and want to know where the blazes you went!!!

Bu varying the allocations of fuel (line 410) you can change the emphasis of the game to one of economy, or to one of extra skill, weaving in and out of the lunar caves, or circumnavigating floating rocks.

Luke has only been an '80 owner for four months, and we congratulate him on his first published effort. He writes that he has many other games of a similar standard... are we interested? Too right, Luke. Send 'em in, please.

```
1 REM LUNALERT / COMMANDER ALERT
2 REM COPYRIGHT (C) 1980 L. BRINDLEY
3 REM                   10 ROSA ST., OATLEY, N.S.W.
4 REM MICRO-80 ISSUE 7, JUNE 1980
5 REM LEVEL I 4K
10 CLS
20 P.A.20,"***COMMANDER ALERT***"
30 F.J=1TO2000:N.J:CLS
40 P.A.140,"COMMUNICATION WITH GROUND CONTROL LOST"
50 P.A.204,"FAILURE OF ONBOARD COMPUTER IMMINIENT"
60 F.J=1TO2000:N.J:CLS
70 F=-1*RND(10)
80 F.I=1TO20
90 P.A.394,"*** ONBOARD COMPUTER HAS MALFUNCTIONED ***"
100 F.J=1TO100:N.J:CLS:N.I
110 P.A.272,"A MANUAL LANDING IS REQUIRED"
120 P.A.340,"INSTRUCTIONS FOLLOW"
130 F.I=1TO2000:N.I:CLS
140 P."ALL ENGINES ARE SHUT DOWN. WE ARE HEADING LEFT AT";F;"KM/H"
150 P."     AND DOWNWARD UNDER THE MOONS GRAVITY AT 0.5 KM/H"
160 P.:P."WE MUST LAND WITH A VERTICLE SPPED OF 1 KM/H OR LESS AND"
170 P." A HORIZONTAL SPEED OF .5 OR LESS IF WE ARE TO SURVIVE."
180 P.:P.
190 P." AT INTERVALS OF 10 SECS. YOU WILL BE REQUIRED TO ENTER"
200 P."              BOTH HORIZONTAL & VERTICLE BURNS."
210 P.:P.:P."        PRESS ENTER TO TAKE CONTROL----GOOD LUCK!"
220 IN.A$
230 CLS:RESTORE
240 P.A.896,"RADAR MAP.    TERRAIN PLOTTED FIRST";
250 F.Y=0TO40:SET(0,Y):N.Y
260 F.X=0TO127:SET(X,40):N.X
270 F.Y=40TO0S.-1:SET(127,Y):N.Y
280 F.X=127TO0S.-1:SET(X,0):N.X
290 F.I=1TO39
300 READA,B,C,D:A=A+RND(A):B=B+RND(35):C=C+RND(A):D=D-RND(127-D)
310 IFA=0G.330
320 F.J=0TOA:SET(J,I):N.J
330 IF(B=0)+(C=0)+(B>C)G.350
340 F.J=BTOC:SET(J,I):N.J
350 IFD=0G.370
360 F.J=DTO127:SET(J,I):N.J
370 N.I
380 D=1000
390 S=.5
400 T=.1
410 W=1000
420 E=0
430 Q=RND(5)
440 P=RND(5)
450 H=120
460 U=3
470 A=H:B=U:K=1:L=5:C=.5:X=1:M=0:G=0:R=0:J=0:V=0
480 P.A.896,"                                                   ";
490 E=F+G*T:H=H+(F*T+G*T*T/2):P=Q+(R+S)*T:U=U+(Q*T+(R+S)*T*T/2):V=V+.5
500 F=E:Q=P:W=W-(ABS(G)*K+ABS(R)*L)
510 P.A.68,"FUEL:";W;"          ";
520 P.A.90,"ELAPSED TIME";V;"          ";
530 P.A.960,"                                             ";
```

```
540 IFW>0G.570
550 P.A.960,"* OUT OF FUEL * FREE FALL FROM HERE ON IN *";
560 M=1:W=0:R=0:G=0
570 IFU<0G.1000
580 IFN=1G.620
590 RESET(A,B)
600 Z=P.(H,U)
610 IFZ=0G.740
620 IF(U<1)+(B<1)G.740
630 IFP<=XG.670
640 P.A.896,"* CRASHED AT";P;"KM/H. YOU HAVE A CRATER NAMED AFTER YOU";
650 P.A.960,"CONGRATULATIONS SIR ON YOUR FINE PERFORMANCE!";
660 G.930
670 IFABS(E)<=CG.710
680 P.A.896,"YOUR VERTICLE IMPACT VELOCITY OF ";P;"WAS ACCEPTABLE   ";
690 P.A.960,"BUT YOU TIPPED OVER AT";P;"KM/H. MAY YOU REST IN PEACE.";
700 G.930
710 P.A.896,"WELCOME TO THE LUNAR SURFACE, CONGRATULATIONS ON A      "
720 P.A.960,"A FINE LANDING. RECEPTION PARTY ON THEIR WAY.";
730 G.930
740 IFU>1T.N=0
750 IFN=1G.770
760 SET(H,U)
770 P.A.896,"VX =";INT(E*100)/100;"    ";
780 P.A.906,"SX =";INT(H*100)/100;"    ";
790 P.A.916,"VY =";INT(P*100)/100;"    ";
800 P.A.926,"SY =";INT(U*100)/100;"    ";
810 P.A.577,"AX =";:P.A.641," ";G;:P.A.646," ";
820 P.A.705,"AY =";:P.A.769," ";R;:P.A.774," ";
830 IFM=1G.900
840 J=J+.5
850 IFJ<10G.900
860 P.A.960,"ENTER X BURN, Y BURN AS:   AX,AY         ";
870 P.A.944,"              ";:P.A.944," ";
880 IN.G,R
890 J=0
900 A=H
910 B=U
920 G.490
930 SET(H,U)
940 F.J=1TO10
950 F.I=1TO100:N.I
960 RESET(H,U)
970 F.I=1TO100:N.I
980 N.J
990 G.10
1000 P.A.960,"OUT OF RADAR RANGE. CONTROL VIA TELEMETRY.";
1010 N=1
1020 G.580
1040 D.1,0,0,127, 2,0,0,126, 2,0,0,125, 2,0,0,125, 2,0,0,125
1050 D.3,0,0,124, 4,0,0,124, 5,0,0,124, 5,0,0,123, 6,0,0,123
1060 D.7,0,0,122, 7,0,0,121, 7,0,0,120, 8,0,0,119, 9,0,0,118
1070 D.10,0,0,118, 11,0,0,118, 12,0,0,117, 13,0,0,117, 14,0,0,117
1080 D.15,0,0,117, 15,0,0,116, 16,0,0,116, 16,0,0,116, 16,0,0,116
1090 D.17,0,0,116, 17,0,0,115, 17,0,0,115, 17,0,0,115, 17,0,0,115
1100 D.18,32,53,114, 18,31,53,113, 18,30,54,112, 19,29,54,112
1110 D.19,28,54,111, 20,28,55,110, 21,28,55,109, 22,27,56,108
1120 D.23,27,57,107, 24,26,58,106
1130 END
```

** ALCOTESCHT ** LEVEL II 16K by A.F. West

As we said last month, a "spoof" program to convince your one-over-the-eight friends to take a taxi home. Very unscientific, but lots of fun, and demonstrating some of the $tring manipulating facilities of the '80.

```
   1 '      ACLOTESCHT

   2 '      MICRO-80, ISSUE 7, JUNE 1980
  10 'BREATH-ALYSER TEST. -   A.F.WEST  13 ROSE ST. BRUNSWICK. 3056
  20 CLS:  FOR I=0TO127: SET(I,0): SET(I,47): NEXT
  30 FOR I=0TO47: SET(0,I): SET(127,I): NEXT
  40 PRINT@210,"***  BREATH-ALYSER TEST  ***";
  50 PRINT@351,"BY";:   PRINT@476,
     "ANNA-LIZA";
  60 PRINT@837,"(I SUGGEST YOU TRY THIS TEST BEFORE YOU DRIVE HOME)";
  70 PRINT@955,"";: INPUT Z
  80 T$="TYPE YOUR SURNAME"
 100 CLS: PRINT
     "TEST #1 ";: INPUT
     "HOW MANY LETTERS IN YOUR SURNAME  ";N
 110 PRINT"TEST #2 ";T$;: A=0
 120 ON Z GOTO 200,300,400,500,600
 129 'COMBINATIONS OF FOUR OF THE 7 OPTIONAL RESULTS
 200 A=0: GOSUB 900: GOSUB 1100
 210 A=1: GOSUB 900: GOSUB 2100: GOSUB 1500
 220 A=2: GOSUB 900: GOSUB 2100: GOSUB 1400
 230 A=3: GOSUB 900: GOSUB 2100: GOSUB 1600
 240 GOTO 2300
 300 A=0: GOSUB 900: GOSUB 1200
 310 A=1: GOSUB 900: GOSUB 2100: GOSUB 1500
 320 A=2: GOSUB 900: GOSUB 2100: GOSUB 1300
 330 A=3: GOSUB 900: GOSUB 2100: GOSUB 1700
 340 GOTO 2200
 400 A=0: GOSUB 900: GOSUB 1300
 410 A=1: GOSUB 900: GOSUB 2100: GOSUB 1200
 420 A=2: GOSUB 900: GOSUB 2100: GOSUB 1400
 430 A=3: GOSUB 900: GOSUB 2100: GOSUB 1600
 440 GOTO 2300
 500 A=0: GOSUB 900: GOSUB 1400
 510 A=1: GOSUB 900: GOSUB 2100: GOSUB 1600
 520 A=2: GOSUB 900: GOSUB 2100: GOSUB 1300
 530 A=3: GOSUB 900: GOSUB 2100: GOSUB 1700
 540 GOTO 2200
 600 A=0: GOSUB 900: GOSUB 1200
 610 A=1: GOSUB 900: GOSUB 2100: GOSUB 1100
 620 A=2: GOSUB 900: GOSUB 2100: GOSUB 1500
 630 A=3: GOSUB 900: GOSUB 2100: GOSUB 1600
 640 GOTO 2300
 899 'TO LOCATE PRINTING ON SCREEN - DEPENDING ON THE ORDER OF RESULTS
     DESIRED
 900 B=170+192*A:  C=192+192*A:  D=64+192*A:  X=15530+192*A:
     RETURN
 910 E=LEN(B$): E$=STRING$(E,32): RETURN
1099 'ERASURE OF NAME
1100 PRINT@B,"";: INPUT B$: GOSUB 910: IF A=0 GOSUB 2050
1110 PRINT@ B+2,E$ : GOSUB 2000
1120 PRINT@C,"YOU'RE MISSING OUT COMPLETELY !": RETURN
```

```
1199 'FIRST LETTERS ERASED
1200 PRINT@B,"";: INPUT B$: GOSUB 910: IF A=0 GOSUB 2050
1210 PRINT@ B+2,STRING$(2,32): GOSUB 2000
1220 PRINT@C,"I WANTED YOUR COMPLETE SURNAME !": RETURN
1299 'LAST LETTERS ERASED
1300 PRINT@B,"";: INPUT B$: GOSUB 910: IF A=0 GOSUB 2050
1310 PRINT@ B+2,E$
1320 PRINT@ B+2,LEFT$(B$,3): GOSUB 2000
1330 PRINT@C,"WHAT ABOUT THE REST ?": RETURN
1399 'MIDDLE LETTERS ONLY PRINTED
1400 PRINT@B,"";: INPUT B$: GOSUB 910: IF A=0 GOSUB 2050
1410 PRINT@ B+2,E$:
1420 PRINT@ B+3,MID$(B$,2,2): GOSUB 2000
1430 PRINT@C,"CAN'T YOU FOCUS PROPERLY !": RETURN
1499 'NAME PRINTED BACKWARDS
1500 POKE X,63:  X=X-2
1510 FOR I=1TO N
1520 A$=INKEY$: IF A$="" THEN 1520
1530 POKE X,ASC(A$)
1540 X=X-1: NEXT: GOSUB 2000
1550 PRINT@C,"BOY - ARE YOU DIS-ORIENTATED !": RETURN
1599 'NAME SLURRED
1600 POKE X,63: X=X+2
1610 FOR I=1TO3: POKE X,95
1620 A$=INKEY$: IF A$="" THEN 1620
1630 POKE X,ASC(A$)
1640 X=X+1: NEXT
1650 POKE X,83: X=X+1: POKE X,72
1660 X=X+1: FOR I=4 TO E: POKE X,95
1670 A$=INKEY$: IF A$="" THEN 1670
1680 POKE X,ASC(A$):  X=X+1:  NEXT:  GOSUB 2000
1689 'SET YOUR OWN STATE LIMIT
1690 PRINT@C,"SHTRUTH YOU'RE PLASHTERED, YOU'RE MILESH ABOVE .05 !!":
     RETURN
1699 'NAME PRINTED IMMEDIATELY
1700 POKE X,63: POKE X+2,95
1710 A$=INKEY$: IF A$="" THEN 1710
1720 PRINT@ B+2,C$
1730 PRINT@C,"THAT'S REALLY ON THE BALL !!": RETURN
2000 FOR I=1TO300: NEXT: RETURN
2050 C$=B$
2060 IF E <> N PRINT"YOU'VE FAILED ALREADY. YOU SAID";N;"LETTERS & YOU
     TYPED";E:  FOR I=1TO2000: NEXT: GOTO 100
2070 RETURN
2100 GOSUB 2000
2110 PRINT@ D,"TRY AGAIN-";TAB(11)T$: RETURN
2200 GOSUB 2000
2210 PRINT@ 896,"I RECKON THAT YOU'RE O.K. TO DRIVE HOME
2220 GOSUB2000: PRINT@1020,"";: INPUT Z
2230 IF Z=0 THEN 2350
2240 GOTO 100
2300 GOSUB 2000
2310 PRINT@896,"ON THOSE RESULTS I RECKON THAT YOU OUGHT TO TAKE A TAXI
     !!"
2320 GOSUB2000: PRINT@1020,"";: INPUT Z
2330 IF Z=0 THEN 2350
2340 GOTO 100
2350 CLS: END
```

** TAPE BLOCKER/UNBLOCKER ** LII any size by B.H. CHRISTENSEN.

Cassette bound '80ers all know the frustrations of using cassette for data storage and retrieval. Each dump takes approximately six seconds, regardless of how much or how little data is being transferred. The effective limit is 248 bytes per transfer dump, and much time can be saved with the technique used in this example.

The data is first converted into $trings and then broken down into "fields" or parcels of identical length. These are then added to the end of a single string buffer until that buffer is full. The buffer is then saved to tape in one hit, thereby minimizing cassette-time wastages.

The first part of the listing (to line 69) is the tape blocker/unblocker itself. The rest demonstrates the use of this routine in a simple cassette-based stock record routine. We all have much to learn from demonstrations of this type, and study of this program is strongly recommended for those with mass storage problems and cassette media.

```
  1 REM TAPE BLOCKER/UNBLOCKER~

    LEVEL II 16K
    MICRO-80 ISSUE 7, JUNE 1980
    AUTHOR B H CHRISTENSEN P.O.BOX 140 WODEN A.C.T. 2606
 10 DEFINTZ: GOTO100
 50 IFZF=1GOTO53
 51 CLS: INPUT
   "MOUNT INPUT CASSETTE  REPLY -GO-";ZR$: IFNOTZR$="GO
   "THENGOTO51ELSEZ1=0:  Z2=0: Z3=0: Z4=0: Z5=0
 52 CLS: POKE15420.76: INPUT#-1,ZB$: POKE15420,32: IFNOTMID$(ZB$.4,4)=
   "HDR1"THENPRINT"NO HEADER LABEL": STOPELSEZ5=LEN(ZB$):
   PRINT"HEADER LABEL",RIGHT$(ZB$,Z5-7): Z5=0: INPUTZR$:
   ZF=1: CLS
 53 IFZ5-Z2>0GOTO57
 54 POKE15420,68: INPUT#-1,ZB$: POKE15420,32: IFNOTMID$(ZB$,4,3)=
   "EOF"THENZ5=VAL(LEFT$(ZB$,3)): Z2=4: Z4=Z4+1: GOTO57
 55 ZT=VAL(MID$(ZB$,7,4)): IFZT=Z3THENINPUT
   "DEMOUNT INPUT CASSETTE";ZR$: ZF=0ELSEPRINT
   "RECORD COUNT OFF    EXPECTED ";ZT;" FOUND ";Z4: STOP:
   ZF=0
 56 INPUT"ANY FURTHER DATA";ZR$: IFZR$="Y"THENGOTO51ELSEIFNOTZR$="N
   "THENGOTO56ELSEZ1=0:  Z2=0: Z3=0: Z4=0: Z5=0: ZB$="":
   CLS: GOTO58
 57 Z1=VAL(MID$(ZB$,Z2,3))-3: ZI$=MID$(ZB$,Z2+3,Z1): Z2=Z2+Z1+3:
   Z3=Z3+1
 58 RETURN
 60 IFZE=1GOTO62
 61 CLS: INPUT
   "MOUNT OUTPUT CASSETTE  REPLY -GO-";ZR$: IFNOTZR$="GO
   "THENGOTO61ELSEINPUT"HEADER LABEL";ZR$: ZS$=STR$(LEN(ZR$)+1007):
   POKE15420,76: PRINT#-1,RIGHT$(ZS$,3)+
   "HDR1"+ZR$: POKE15420,32: Z7=0: Z8=0: Z9=0: ZE=1: ZA$=
   ""
 62 Z6=LEN(ZO$)+3: IFZ7+Z6>245GOSUB64
 63 ZA$=ZA$+RIGHT$(STR$(Z6+1000),3)+ZO$: Z7=Z7+Z6: Z8=Z8+1:
   GOTO69
 64 : POKE15420,68: PRINT#-1,RIGHT$(STR$(Z7+1003),3)+ZA$:
   POKE15420,32: Z9=Z9+1: Z7=0: ZA$="": RETURN
 65 IFZ7>0GOSUB64
 66 Z8$=STR$(Z8+10000): Z9$=STR$(Z9+10000): POKE15420,76:
   PRINT#-1,"014EOF"+RIGHT$(Z8$,4)+RIGHT$(Z9$,4): POKE15420,32:
   Z7=0: Z8=0: Z9=0: INPUT
   "DEMOUNT OUTPUT CASSSETTE";ZR$: ZE=0
 69 RETURN
 98 '***** ***** ***** ***** ***** ***** ***** ***** ***** ***** *****
    *****
 99 REM THIS IS A SAMPLE PROGRAM TO ILLUSTRATE THE USE OF TAPE BLOCKER/
    UNBLOCKER   IT WRITES SOME SIMPLE STOCKTAKE RECORDS TO CASSETTE
    THEN READS THEM BACK AND   PRINTS THEM
100 CLEAR750
110 CLS: PRINT
    "PLEASE ENTER STOCK DETAILS AS REQUESTED": INPUT
    "PART NO. (6 DIGITS)";PN$: IFNOTLEN(PN$)=6THEN110
120 INPUT"DESCRIPTION (UP TO 80 CHARACTERS)";DE$
130 INPUT"UNIT COST (NOT OVER $9999.99)";UC$: UC$=RIGHT$(
    "00000"+UC$,7): REM CONVERTED TO FIXED LENGTH 7 BYTE FIELD WITH
    LEADING ZEROS
140 INPUT"NUMBER ON HAND (NOT OVER 9999)";OH: OH$=RIGHT$(STR$(10000+OH),
    4):  REM CONVERTED TO 4 BYTE STRING FIELD
150 ZO$=PN$+UC$+OH$+DE$: REM SET UP RECORD WITH ALL FIELDS FIXED LENGTH
    EXCEPT THE LAST
160 GOSUB60:  REM CALL TAPE BLOCKER/UNBLOCKER
170 INPUT"ANY FURTHER RECORDS (Y OR N)";R$: IFR$="Y
    "THENGOTO110ELSEIFNOTR$="N"THENGOTO170
180 GOSUB65: REM CALL TAPE BLOCKER/UNBLOCKER TO WRITE LAST BLOCK AND
    CLOSE FILE
200 REM PART 2   READ RECORDS FROM CASSETTE AND PRINT
210 PF$="$$###.##": GOSUB50: REM CALL TAPE BLOCKER/UNBLOCKER
220 IFZF=0THENPRINT"TOTAL RECORDS = ";RC: ENDELSERC=RC+1:
     REM TEST FLAG FOR END OF FILE
230 T=LEN(ZI$): DE$=RIGHT$(ZI$,T-17)
240 PN$=LEFT$(ZI$,6)
250 UC=VAL(MID$(ZI$,7,7))
260 OH=VAL(MID$(ZI$,14,4))
265 IFRC=1THENCLS: PRINT
    "PART NO       DESCRIPTION" TAB(45)"UNIT COST   ON HAND"
270 PRINT PN$ TAB(10)DE$ TAB(45)USINGPF$;UC;: PRINTTAB(57)OH
280 GOTO210
```

** ACEY DUCEY ** LEVEL II 16K by C. Parkinson

The basis of "Acey Ducey" is to bet on whether a third card dealt from a random pack will lie within the values of the first two cards so dealt. Sounds simple enough, and so it is... Becuase this version uses random selections with each hand, it really presents a variety of Acey Ducey where the pack is reshuffled after each hand, which is not the usual method of play. Some of you may like to create an array that is shuffled at the start of play, always being dealt from the top.

Nevertheless, this game is worthy of adding to your libraries, if only because it's a nice change to see a "different" game once in a while.

This is a one or two player game, each starting with a bank of $100, and is a very good teacher of the "odds" for those wanting some quite practise at home before "conning" one of their mates into being "banker" for the night.

```
  1 ' ACEY DUCEY
  2 ' |EVEL ii 16k
  3 ' MICRO-8Ø ISSUE 7, JUNE 198Ø
 1Ø '** ACEY DUCEY. BY C.PARKINSON 47 SILLMON WAY DUNCRAIG (PERTH) W.A.
    6Ø23 **
 2Ø DEFINTA-Z: CLS: PRINT@23,
    "* * ACEY-DUCEY * *": PRINT: PRINT: PRINT: PRINT: INPUT
    "DO YOU WANT THE RULES";R$: IFR$=
    "YES"GOTO54Ø
 3Ø IFR$="NO"GOTO5Ø
 4Ø PRINT@453,"PLEASE ANSWER YES OR NO": FORT=1TO1ØØØ: NEXT:
    GOTO2Ø
 5Ø CLS: INPUT
    "1 OR 2 PLAYERS";P: IFP=2GOTO57Ø
 6Ø IFP<1ORP>2THENPRINT"DON'T BE SILLY": FORT=1TO2ØØØ :  NEXT:
     GOTO 5Ø
 7Ø INPUT"WHAT IS YOUR NAME";P1$
 8Ø T1=1ØØ: P2=Ø
 9Ø CLS: IFT1=ØGOTO51Ø
1ØØ RANDOM: A=INT(RND(13))
11Ø B=INT(RND(13)): IFA=BTHEN11Ø
12Ø S1=INT(RND(4)): S2=INT(RND(4))
13Ø C=INT(RND(13)): S3=INT(RND(4)): IFC=AANDS3=S1THEN13Ø
14Ø IFC=BANDS3=S2THEN13Ø
15Ø FORX=12TO35: Y=3: SET(X,Y): NEXT: FORX=12TO35: Y=16: SET(X,Y):
    NEXT: FORY=3TO16: X=12: SET(X,Y): NEXT: FORY=3TO16: X=35:
    SET(X,Y): NEXT: FORX=52TO75: Y=3: SET(X,Y): NEXT: FORX=52TO75:
    Y=16: SET(X,Y): NEXT: FORY=3TO16: X=52: SET(X,Y): NEXT
16Ø FORY=3TO16: X=75: SET(X,Y): NEXT: IFA=1PRINT@138,
    "ACE";: GOTO21Ø
17Ø IFA=11PRINT@138,"JACK";: GOTO21Ø
18Ø IFA=12PRINT@137,"QUEEN";: GOTO21Ø
19Ø IFA=13PRINT@138,"KING";: GOTO21Ø
2ØØ PRINT@139,A;
21Ø REM*PRINT CARD VALUES *: IFB=1PRINT@159,
    "ACE";: GOTO5ØØ
22Ø IFB=11PRINT@158,"JACK";: GOTO26Ø
23Ø IFB=12PRINT@157,"QUEEN";: GOTO26Ø
24Ø IFB=13PRINT@158,"KING";: GOTO26Ø
25Ø PRINT@159,B;
26Ø 'DUMY: PRINT@17Ø,;P1$;
    "'S CARDS";: IFS1=1PRINT@199,
    "OF HEARTS";: GOTO55Ø
27Ø IFS1=2PRINT@199,"OF CLUBS";: GOTO3ØØ
28Ø IFS1=3PRINT@199,"OF SPADES";: GOTO3ØØ
29Ø IFS1=4PRINT@199,"ØF DIAM'S";: GOTO3ØØ
3ØØ 'DUMY: IFS2=1PRINT@219,
    "OF HEARTS";
31Ø IFS2=2PRINT@219,"OF CLUBS";
32Ø IFS2=3PRINT@219,"OF DIAM'S";
33Ø IFS2=4PRINT@219,"OF SPADES";
34Ø 'DUMY: FORX=32TO55: Y=21: SET(X,Y): NEXT: FORX=32TO55:
    Y=33: SET(X,Y): NEXT: FORY=21TO33: X=32: SET(X,Y): NEXT:
    FORY=21TO33: X=55: SET(X,Y): NEXT: PRINT@533,"MY";: PRINT@596,
    "CARD";
```

```
350 PRINT@832,P1$;" YOU NOW HAVE $"T1;: INPUT
    "HOW MUCH DO YOU WANT TO BET";B1: IFB1>T1THENPRINT@896,
    "SORRY YOU HAVEN'T GOT THAT MUCH"
360 FORT=1TO500: NEXT: PRINT@896,CHR$(30);: PRINT@832,CHR$(30);:
    IFB1>T1THEN350
370 PRINT@533,"  ";: PRINT@596,
    "     ";: IFC=1PRINT@532,
    "ACE";: GOTO420
380 IFC=11PRINT@531,"JACK";: GOTO420
390 IFC=12PRINT@530,"QUEEN";: GOTO420
400 IFC=13PRINT@531,"KING";: GOTO420
410 PRINT@533,;C;
420 IFS3=1PRINT@593,"OF HEARTS";: GOTO460
430 IFS3=2PRINT@593,"OF CLUBS";: GOTO460
440 IFS3=3PRINT@593,"OF DIA'S";: GOTO460
450 IFS3=4PRINT@593,"OF SPADES";
460 FORT=1TO2000: NEXT: IFA>BTHEN480
470 IFB>ATHEN490
480 IFC<A ANDC>BTHENPRINT@832,"YOU WIN $";B1: T1=T1+B1: FORT=1TO2000:
    NEXT: GOTO90
490 IFC<B ANDC>ATHENPRINT@832,"YOU WIN $";B1: T1=T1+B1: FORT=1TO2000:
    NEXT: GOTO90
500 PRINT@832,"SORRY YOU LOSE": T1=T1-B1: FORT=1TO2000: NEXT:
    GOTO90
510 CLS: PRINT: PRINT: PRINT: PRINT
    "WELL "P1$;" YOU HAVE LOST ALL YOUR MONEY": PRINT
    "DO YOU WANT TO PLAY AGAIN": INPUTA$: IFA$=
    "YES"ANDP2=0GOTO80
520 IFA$="YES"ANDP2=1GOTO570
530 CLS: PRINT: PRINT: PRINT: PRINT: PRINT
    "O.K. HOPE YOU HAD FUN": FORT=1TO2000: NEXT: GOTO20: PRINT@768,
    "STOP": STOP
540 CLS: PRINT: PRINT: PRINT: PRINT
    "ACEY-DUCEY IS PLAYED BY 1 OR 2 PLAYERS": PRINT
    "THE COMPUTER WILL DEAL YOU 2 CARDS FACE UP, YOU HAVE THE OPTION":
    PRINT"OF BETTING ON THE NEXT CARD BEING BETWEEN YOUR 2 CARDS"
550 PRINT"IF THE NEXT CARD IS EQUAL TO EITHER OF YOUR CARDS, YOU LOSE"
560 FORZ=591TO626STEP2: PRINT@Z,"*": FORT=1TO300: NEXTT,Z:
    GOTO50
570 CLS: PRINT: PRINT: INPUT
    "FIRST PLAYERS NAME ";P1$: T1=100: T2=100: P2=1: INPUT
    "SECOND PLAYERS NAME";P2$
580 CLS: FORX=12TO35: Y=0: SET(X,Y): NEXT: FORX=12TO35: Y=12:
    SET(X,Y): NEXT: FORY=0TO12: X=12: SET(X,Y): NEXT: FORY=0TO12:
    X=35: SET(X,Y): NEXT: FORX=52TO75: Y=0: SET(X,Y): NEXT:
    FORX=52TO75: Y=12: SET(X,Y): NEXT: FORY=0TO12: X=52: SET(X,Y):
    NEXT
590 FORY=0TO12: X=75: SET(X,Y): : NEXT
600 A1=INT(RND(13)): B1=INT(RND(13))
610 A2=INT(RND(13))
620 B2=INT(RND(13))
630 C=INT(RND(13)): S1=INT(RND(4)): S2=INT(RND(4)): S3=INT(RND(4)):
    S4=INT(RND(4)): S5=INT(RND(4)): IFA1=B1ANDS1=S2THEN600
640 IFA1=A2ANDS1=S3THEN610
650 IFA1=B2ANDS1=S4THEN620
660 IFA1=CANDS1=S5THEN630
670 IFB1=A2ANDS2=S3THEN610
680 IFB1=B2ANDS2=S4THEN620
690 IFB1=CANDS2=S5THEN630
700 IFA2=B2ANDS3=S4THEN620
710 IFA2=CANDS3=S5THEN630
720 IFB2=CANDS4=S5THEN630
730 IFA1=1PRINT@74,"ACE";: GOTO780
```

```
740 IFA1=11PRINT@74,"JACK";: GOTO780
750 IFA1=12PRINT@73,"QUEEN";: GOTO780
760 IFA1=13PRINT@74,"KING";: GOTO780
770 PRINT@75,A1;
780 IFS1=1PRINT@135,"OF HEARTS";: GOTO820
790 IFS1=2PRINT@135,"OF CLUBS";: GOTO820
800 IFS1=3PRINT@135,"OF DIA'S";: GOTO820
810 IFS1=4PRINT@135,"OF SPADES";
820 IFB1=1PRINT@94,"ACE";: GOTO870
830 IFB1=11PRINT@93,"JACK";: GOTO870
840 IFB1=12PRINT@92,"QUEEN";: GOTO870
850 IFB1=13PRINT@93,"KING";: GOTO870
860 PRINT@95,B1;
870 IFS2=1PRINT@155,"OF HEARTS";: GOTO910
880 IFS2=2PRINT@155,"OF CLUBS";: GOTO910
890 IFS2=3PRINT@155,"OF DIA'S";: GOTO910
900 IFS2=4PRINT@155,"OF SPADES";
910 FORX=12TO35: Y=18: SET(X,Y): NEXT: FORX=52TO75: SET(X,Y):
    NEXT: FORX=12TO35: Y=30: SET(X,Y): NEXT: FORX=52TO75:
    SET(X,Y): NEXT: FORY=18TO30: X=12: SET(X,Y): NEXT: FORY=18TO30:
    X=35: SET(X,Y): NEXT: FORY=18TO30: X=52: SET(X,Y): NEXT
920 FORY=18TO30: X=75: SET(X,Y): NEXT: IFA2=1PRINT@458,
    "ACE";: GOTO970
930 IFA2=11PRINT@457,"JACK";: GOTO970
940 IFA2=12PRINT@456,"QUEEN";: GOTO970
950 IFA2=13PRINT@457,"KING";: GOTO970
960 PRINT@459,A2;
970 IFS3=1PRINT@519,"OF HEARTS";: GOTO1010
980 IFS3=2PRINT@519,"OF CLUBS";: GOTO1010
990 IFS3=3PRINT@519,"OF DIA'S";: GOTO1010
1000 IFS3=4PRINT@519,"OF SPADES";
1010 IFB2=1PRINT@478,"ACE";: GOTO1060
1020 IFB2=11PRINT@477,"JACK";: GOTO1060
1030 IFB2=12PRINT@476,"QUEEN";: GOTO1060
1040 IFB2=13PRINT@477,"KING";: GOTO1060
1050 PRINT@479,B2;
1060 IFS4=1PRINT@539,"OF HEARTS";: GOTO1100
1070 IFS4=2PRINT@539,"OF CLUBS";: GOTO1100
1080 IFS4=3PRINT@539,"OF DIA'S";: GOTO1100
1090 IFS4=4PRINT@539,"OF SPADES";
1100 PRINT@104,P1$;"'S CARDS";: PRINT@552,P2$;
     "'S CARDS";: FORX=82TO105: Y=7: SET(X,Y): NEXT: FORX=82TO105:
     Y=21: SET(X,Y): NEXT: FORY=7TO21: X=82: SET(X,Y): NEXT:
     FORY=7TO21: X=105: SET(X,Y): NEXT: PRINT@302,"MY";: PRINT@365,
     "CARD";
1110 PRINT@768,P1$;" YOU NOW HAVE $";T1;: INPUT
     "HOW MUCH DO YOU WANT TO BET";W1: IFW1<0PRINT
     "DON'T BE SILLY": FORT=1TO1000: NEXT: GOTO1140
1120 IFW1<=T1THEN1150
1130 PRINT"YOU DON't HAVE THAT MUCH": FORT=1TO1000: NEXT
1140 PRINT@768,CHR$(30): PRINT
     "                                    ": GOTO1110
1150 PRINT@832,P2$;" YOU NOW HAVE $";T2;: INPUT
     "HOW MUCH DO YOU WANT TO BET";W2: IFW2<0PRINT
     "DON'T BE SILLY": FORT=1TO1000: NEXT: GOTO1180
1160 IFW2<=T2THEN1190
1170 PRINT"YOU DON'T HAVE THAT MUCH": FORT=1TO1000: NEXT
1180 PRINT@832,CHR$(30): PRINT
     "                                             ": GOTO1150
```

```
1190 IFC=1PRINT@301,"ACE";: GOTO1240
1200 IFC=11PRINT@300,"JACK";: GOTO1240
1210 IFC=12PRINT@299,"QUEEN";: GOTO1240
1220 IFC=13PRINT@300,"KING";: GOTO1240
1230 PRINT@302,C;
1240 IFS5=1PRINT@362,"OF HEARTS";: GOTO1280
1250 IFS5=2PRINT@362,"OF CLUBS";: GOTO1280
1260 IFS5=3PRINT@362,"OF DIA'S";: GOTO1280
1270 IFS5=4PRINT@363,"OF SPADES";
1280 FORT=1TO1000: NEXT: PRINT@704,"": PRINT@768,"": IFA1>B1THEN1370
1290 IFC>A1ANDC<B1PRINT@800,P1$;" WINS $";W1;: T1=T1+W1: GOTO1310
1300 GOTO1380
1310 IFA2>B2THEN1390
1320 IFC>A2ANDC<B2PRINT@864,P2$;" WINS $";W2;: T2=T2+W2: GOTO1340
1330 GOTO1400
1340 FORT=1TO2000: NEXT: IFT1=0THEN1410
1350 IFT2=0THEN1430
1360 GOTO580
1370 IFC>B1ANDC<A1PRINT@800,P1$;" WINS $";W1;: T1=T1+W1: GOTO1310
1380 PRINT@800,P1$;" LOSES $";W1;: T1=T1-W1: GOTO1310
1390 IFC>B2ANDC<A2PRINT@864,P2$;" WINS $";W2;: T2=T2+W2: GOTO1340
1400 PRINT@864,P2$;" LOSES $";W2;;: T2=T2-W2: GOTO1340
1410 IFT2=0THEN1440
1420 CLS: PRINT@460,
     "SORRY ";P1$;" BUT YOU ARE BROKE": PRINTTAB(10);P2$;
     " YOU MAY CONTINUE": FORT=1TO5000: NEXT: P1$=P2$: T1=T2:
     GOTO90
1430 CLS: PRINT@460,
     "SORRY ";P2$;" BUT YOU ARE BROKE": PRINTTAB(10);P1$;
     " YOU MAY CONTINUE": FORT=1TO5000: NEXT: P2=0: GOTO90
1440 CLS: PRINT@460,
     "SORRY BUT YOU ARE BOTH BROKE": PRINT: PRINT: PRINT: INPUT
     "AGAIN";A$: IFA$=
     "YES"THEN50
1450 CLS: PRINT: PRINT: PRINT: PRINT
     "O.K. HOPE YOU HAD FUN": PRINT
     "BYE BYE": FORT=1TO1000: NEXT: CLS: END
```

** GRAPHIX ** LII/16K m.l.

This program produces fascinating moving graphics displays. The program will ask values for X, Y, type and speed. The limits are as follows :

X 1-127 Limit for horizontal direction.
Y 1-47 Limit for vertical direction.
TYPE 0-7 Options (0-3: 4 Corner 4-7: 1 Corner).
SPEED 0-255 Speed of display (0 = fastest).
If don't want to change one value, hit the ENTER key instead.
To stop the display and enter new values, hit the BREAK key.

```
7000  F3 21 0F 72 CD 4D 71 CD A3 71 21 4D 72 CD 4D 71
7010  21 BB 72 0E 05 CD 57 71 21 BB 72 7E FE 0D 28 0E
7020  CD 86 71 B7 28 E4 C6 00 FA 0A 70 32 0C 72 CD A3
7030  71 21 63 72 CD 4D 71 21 BB 72 0E 05 CD 57 71 21
7040  BB 72 7E FE 0D 28 0E CD 86 71 B7 28 E4 FE 30 F2
7050  31 70 32 0D 72 CD A3 71 21 89 72 CD 4D 71 21 BB
7060  72 0E 05 CD 57 71 21 BB 72 7E FE 0D 28 06 CD 86
7070  71 32 0E 72 CD A3 71 21 79 72 CD 4D 71 21 BB 72
7080  0E 05 CD 57 71 21 BB 72 7E FE 0D 28 06 CD 86 71
7090  32 BA 72 21 00 3C 11 01 3C 01 00 04 36 80 ED B0
70A0  DD 21 08 72 3E 01 21 08 72 06 04 77 23 10 FC DD
70B0  7E 00 DD 86 02 DD 77 00 DD 7E 01 DD 86 03 DD 77
70C0  01 DD 7E 04 DD BE 00 30 16 DD 7E 02 2F DD CB 06
70D0  4E 20 01 3C DD 77 02 CB 27 DD 86 00 DD 77 00 DD
70E0  7E 00 E6 80 20 E3 DD 7E 05 DD BE 01 30 14 DD 7E
70F0  03 2F DD CB 06 46 20 01 3C DD 77 03 DD 86 01 DD
7100  77 01 DD 7E 01 E6 80 20 E5 DD 56 00 DD 5E 01 CD
7110  D8 71 DD CB 06 56 20 14 3E 7F 92 57 CD D8 71 3E
7120  2F 93 5F CD D8 71 DD 56 00 CD D8 71 3A BA 72 B7
7130  28 08 47 3E FF 3D 20 FD 10 F9 3A 40 38 E6 04 CA
7140  AF 70 21 FF FF 2B 7C B5 20 FB C3 00 70 7E FE 03
7150  C8 CD 33 00 23 18 F6 06 00 3E 0E CD 33 00 AF CD
7160  2B 00 B7 28 F9 77 FE 0D C8 FE 08 28 0D 78 3C B9
7170  30 EC 47 7E CD 33 00 23 18 E4 78 3D 38 E0 47 7E
7180  CD 33 00 2B 18 D8 06 00 7E D6 30 38 14 FE 0A 30
7190  10 4F 78 CB 27 CB 27 CB 27 CB 20 80 81 47 23 18
71A0  E7 78 C9 26 00 3A 0C 72 6F DD 21 9F 72 CD C0 72
71B0  3A 0D 72 6F DD 21 A6 72 CD C0 72 3A 0E 72 6F DD
71C0  21 AD 72 CD C0 72 3A BA 72 6F DD 21 B4 72 CD C0
71D0  72 21 98 72 CD 4D 71 C9 7B 26 FF 24 D6 03 30 FB
71E0  C6 03 44 4A CB 21 CB 18 CB 19 CB 18 CB 19 17 3C
71F0  37 26 00 CB 14 3D 20 FB 78 F6 FC E6 3F 47 0A AC
7200  F6 80 02 C9 53 54 4F 52 47 23 FF FF 27 28 03 1C
7210  1F 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
7220  20 20 20 20 20 0D 0D 50 52 45 53 53 20 42 52 45
7230  41 4B 20 54 4F 20 52 45 54 55 52 4E 20 54 4F 20
7240  54 48 49 53 20 44 49 53 50 4C 41 59 03 1D 1E 45
7250  4E 54 45 52 20 4D 41 58 20 58 20 56 41 4C 55 45
7260  3A 20 03 1D 1E 45 4E 54 45 52 20 4D 41 58 20 59
7270  20 56 41 4C 55 45 3A 20 03 1D 1E 45 4E 54 45 52
7280  20 53 50 45 45 44 3A 20 03 1D 1E 45 4E 54 45 52
7290  20 54 59 50 45 3A 20 03 1C 1A 1A 1A 1A 58 3D 31
72A0  30 30 20 20 59 3D 20 34 30 20 20 54 3D 20 20 30
72B0  20 20 53 3D 20 20 30 0D 0D 03 00 0D 32 37 0D 08
72C0  0E 20 06 03 FD 21 F4 72 FD 5E 00 FD 23 FD 56 00
72D0  FD 23 97 ED 52 DA DC 72 3C C3 D3 72 19 B7 CA E3
72E0  72 0E 30 B1 DD 77 00 DD 23 10 DD DD 7E FF F6 30
72F0  DD 77 FF C9 64 00 0A 00 01 00 20 44 49 53 50 4C
7300  41 40 40 40 40 40 40 40 40 40 40 40 40 40 40 40
```

***** NEXT MONTH'S ISSUE *****

Chess games take a long time to play and the Chess War article was not quite completed by press-time but should be ready for next month's issue. Pressure of space has also forced us to hold over Part 5 of Assembly Language Programming until the July issue. Incidentally, you can expect the July issue to come out much earlier in the month than recent issues. It is our aim to mail out copies in the first week of each month and we should go a long way towards achieving this from next month on.
The July issue of MICRO-80 will contain at least the following:-

CATCH (L1) A simple but fascinating graphics-based game that will entertain for hours

SYMBOL/BAS(L2). Our first venture into the arena of educational programs. This example ties in closely with Spencer George's article elsewhere in this edition.

INTEREST(L2) One to give our Banker and Accountant friends something to ponder over. The Author claims that the approach he has adopted is original and it has certainly left us scratching our heads.

FUEL ECONOMY(L1). With the rising price of "go-juice" and the pure confusion of metrication, this one is a must for any cost conscious motorist

SOUND-OFF (m.l.) Two programs written by readers who have been getting stuck into some m.l. square-ware generation. We expect that these examples will be the basis for a regular spot for noisy MICRO-80 readers.

Plus Part 5 of Assembly Language Programming and all the usual articles, letters, news etc.

tick where appropriate

To MICRO-80
Please consider the enclosed program for...

(i) Publication in MICRO-80

(ii) Publication on disk or cassette only

(iii) Both

Name...

Address..

....

......................Post Code.......................

***** CHECK LIST *****

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes seperated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it...etc

For system tapes, the start, end, and entry points, etc

The changes or improvements that you think may improve it.

Please package securely - padabags are suggested - and enclose stamps or postage if you want your cassette or disk returned.

***** CASSETTE EDITION INDEX *****

Level I programs are on Side 1; Level II programs are on Side 2. Each program is recorded twice in succession.

		I.D.	APPROX. START POSITION CTR-41	CTR-80
SIDE 1				
Beethoven's Helper	L1/4K	–	16	10
			97	60
Lunalert	L1/4K	–	170	105
			234	145
SIDE 2				
Tapeblocker/Unblocker	L2/4K	T	8	5
			55	34
Alcotescht	L2/4K	B	87	54
			127	79
Acey Ducey	L2/16K	A	158	98
			218	135
Graphix	L2/16K	SYSTEM	282	175
		"GRAPHX"	290	180

Individual copies of the Issue 7 cassette may be purchased by SUBSCRIBERS for $3.50. The combined magazine and cassette subscription rate is $60.00 for 12 issues. Why risk missing out? Take out a subscription today.

To: MICRO-80, P.O. Box 213, Goodwood, S.A. 5034.

Please rush me the items checked below:

...12 month subscription to MICRO-80 $24.00

...12 month subscription to MICRO-80 plus the cassette edition $60.00

...The latest issue of MICRO-80 $ 2.50

The MICRO-80 PRODUCTS listed below:

DESCRIPTION	PRICE
TOTAL ENCLOSED WITH ORDER	

□ Cheque □ Bankcard □ Money Order

Bankcard Account Number □□□□□□□□□□□□□

Signature..........................Exp. End..............

NAME...

ADDRESS...............................POST CODE..........

A SPECIAL OFFER!

TO

SUBSCRIBERS OF **MICRO 80**

FOR THE FIRST TIME IN AUSTRALIA!
GENUINE TANDY TRS–80 MICRO-COMPUTERS, HARDWARE AND SOFTWARE AT

GREAT DISCOUNT PRICES!!!

HOW TO ORDER –

1. SELECT ITEMS FROM 1980 TANDY CATALOGUE
2. DEDUCT 10% FROM ADVERTISED PRICES
3. POST US YOUR ORDER STATING DESCRIPTION CAT. No AND A CHEQUE OR MONEY ORDER.

WE WILL –

1. ATTEND TO YOUR ORDER WITHIN 7 DAYS *
2. SUPPLY GOODS SELECTED FREIGHT FREE !
3. SEND ADVERTISING REGULARLY TO KEEP YOU INFORMED OF CURRENT SPECIALS !

** subject to availability*

to **CONQUEST ELECTRONICS** Pty. Ltd.
212 Katoomba St. KATOOMBA 2780

Please supply –

QTY.	CAT NO	DESC.	ADV. PRICE

SUB TOTAL ....
LESS 10% ....
FIND CHEQUE FOR **TOTAL** ....

SEND **FREIGHT FREE** TO
NAME ....
ADDRESS ....
.... P/ Code ....

TANDY
ELECTRONICS
DEALER

CONQUEST ELECTRONICS PTY. LTD.

212 KATOOMBA ST KATOOMBA N.S.W. 2780 PHONE (047) 82 2491

MICRO-80

LEVEL II ROM REFERENCE MANUAL

by Edwin Paay

Published by MICRO-80 PRODUCTS

Written by Eddy Paay, the LEVEL II ROM REFERENCE MANUAL is the most complete explanation of the Level II BASIC interpreter ever published.

Part 1 lists all the useful and usable ROM routines, describes their functions explains how to use them in your own machine language programs and notes the effect of each on the various Z 80 registers.

Part 1 also details the contents of system RAM and shows you how to intercept BASIC routines as they pass through system RAM. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory—the only restriction is your own imagination!

Part 2 gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements, etc. It also describes the various formats used for BASIC, SYSTEM and EDITOR/ASSEMBLER tapes. Each section is illustrated by sample programs which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

The LEVEL II ROM REFERENCE MANUAL is intended to be used by machine language programmers. It assumes a basic understanding of the Z 80 instruction set and some experience of Assembly Language programming. But BASIC programmers too will benefit from reading it. They will gain a much better insight into the functioning of the interpreter which should help them to write faster, more concise BASIC programs.

MICRO-80